VAX-11/750

EK-GA75Ø-RM-ØØ1

VAX-11/75Ø
GATE ARRAY CHIP
REFERENCE MANUAL

The material in this manual is for informational purposes and is subject to change without notice

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual

Printed in U S A

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DIGITAL
DEC
PDP
DECUS
UNIBUS

DECsystem-10
DECSYSTEM-20
DIBOL
EDUSYSTEM
VAX
VMS

MASSBUS
OMNIBUS
OS/8
RSTS
RSX
IAS

CONTENTS

1. DOCUMENT SCOPE

The Gate Array Chip Reference is Company Confidential. It is available only on fiche. The intent of this document is to provide a detailed information source for gate array chips used in the VAX-11/750 Central Processor, UNIBUS Interface, and MASSBUS Adaptor. The information contained in this manual should be read in reference to the Field Service 11/750 Gate Array Print Set (MP-XXX) see Note.

Note: XXX = Number to be supplied at a later date.

Table 1-1 Related Manuals

Title	Document Number
Technical Descriptions:	
DW750 Unibus Interface (UBI)	EK-DW750-TD
MS750 Memory System	EK-MS750-TD
PS750 Power System	EK-PS750-TD
RH750 Massbus Adapter (MBA)	EK-RH750-TD
FP750 Floating-Point Accelerator (FPA)	EK-FP750-TD
Diagnostic System:	
VAX-11 Diagnostic System User's Guide	EK-VX11D-UG
VAX-11/750 Diagnostic System Overview	EK-VXD75-UG
User Documentation:	
Site Preparation Data Sheets	EK-CORP-SP
Installation/Acceptance	EK-SI750-IN
VAX-11 Architecture Handbook	EB-17580-18
VAX-11 Software Handbook	EB-15485-18
VAX-11 Hardware Handbook	EB-17281-20
Field Service 11/750 Gate Array Print Set	MP-XXX

Hard-copy documents can be ordered from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532
Attn: Printing and Circulation Services (NR2/M15)

For information concerning microfiche libraries, contact:

Digital Equipment Corporation
Micropublishing Group
26 Crosby Drive
Bedford, MA 01730

2. GATE ARRAY TECHNOLOGY (General Overview)

The majority (90%) of the VAX-11/750 hardware logic design is implemented in custom large scale integrated (LSI) circuits called gate arrays. These gate arrays are designed and manufactured specifically for the VAX-11/750. Gate array technology utilizes a fixed physical placement of 400 NAND gates (these gates are comprised of bipolar circuit technology). Each gate array chip is configured during the manufacturing process to produce one of the 39 different gate array types used in the VAX-11/750. These chips are used in the VAX-11/750 CPU, Floating Point accelerator, Memory Controller and Mass Bus adaptor.

Custom gate array technology has produced a positive impact on the VAX-11/750 design in a number ways:

- Increased speed per logic gate (5 to 10 nanoseconds)
- Lowered power consumption
- Fewer P.C. boards due to LSI
- Increased reliability
- Lowered cost

For details on the preceding points see Table 2-1.

Table 2-1 Technology Specifications for the VAX-11/750

Implementation Technique - Gate Arrays
Circuit Technology - Low Power Bipolar Schottky
Circuit Density - Large Scale Integration (LSI)

Die Size - .215 inches x .244 inches
Power Utilized per Die - 2 watts Max

Package Size - 1.44 sq. in. (2.4 inches x 0.6 inches)
Number of Pins/Package - 48

I/O Circuits/die - 44 I/O transceiver gates
Logic Gates - 400 identical 4 input NAND gates

Voltage Used - +2.5 volts, + 5 volts
Speed per gate - 4-10 nanoseconds

Unique Gate Array Types:

CPU and Memory Controller - 27
Floating Point Accelerator - 7
Mass Bus Adaptor - 5

Total Number of Gate Arrays Used:

DPU and Memory Controller - 55
Floating Point Accelerator - 28
Mass Bus Adaptor - 12

3. GATE ARRAY CHIP DATA

3.1 ACCESS CONTROL CHIP (ACV-DC625)

1. GENERAL DESCRIPTION:

The ACV is a control chip which detects and prioritizes several kinds of micro-traps and encodes the identity of the highest priority micro-trap on three pins for use by UTR. It also provides two bits of micro-vector information on PTE checks and ACCESS PROBES.

ACV Table 1

Pin Identification and Gate Type

			GATE
PIN #		PIN	
ID	COMMENTS	TYPE	
1	BUS 4 H		GA1TNF
2	M CLK ENABLE H		GA1TNF
3	WBUS 24 H		GA1TNF
4	AC 0 H		GA1TNF
5	AC 3 H		GA1TNF
6	AC 1 H		GA1TNF
7	AC 2 H		GA1TNF
8	UTRAP L		GA1TNF
9	ACV H		GA1TNF
10	LATCHED BUS 0 H		GA1TNF
11	LATCHED BUS 3 H		GA1TNF
14	TB VALID H		GA1TNF
15	PTE CHECK OR PROBE H		GA1TTN
16	MICRO VECTOR 0 H		GA1TZN
17	MICRO VECTOR 1 H		GA1TZN
18	ENC UTRAP 1 L		GA1TTN
19	ENC UTRAP 0 L		GA1TTN
20	UB ADDRESS H		GA1TNH
21	ENC UTRAP2 L		GA1TTN
22	PREFETCH L		GA1TNF
23	CS PARITY ERROR H		GA1TNF
24	LATCHED BUS		GA1TNF
25	FP RES OP L		GA1TNF
26	D SIZE 1 H		GA1TNF
27	LATCHED WCTRL 4 H		GA1TNF
28	LATCHED WCTRL 2 H		GA1TNF
29	MA 02 H		GA1TNF
30	D CLK ENABLE H		GA1TNF
31	LATCHED WCTRL 5H		GA1TNF
32	MA 01 H		GA1TNF
33	LATCHED WCTRL 1 H		GA1TNF
34	MA 00 H		GA1TNF
36	LATCHED WCTRL 3 H		GA1TNF
37	LATCHED BUS 1 H		GA1TNF
39	LATCHED WCTRL 0 H		GA1TNF
40	FORCE MA 09 H		GA1TNN
41	PHASE 1 H		GA1TNF
42	WBUS 26 H		GA1TNF
43	WBUS 27 H		GA1TNF
44	WBUS 25 H		GA1TNF
45	PAGE BNDRY H		GA1TNF
46	PROC INIT H		GA1TTN
47	B CLK L		GA1TNF
48	D SIZE 0 H		GA1TNF

2. PERFORMANCE REQUIREMENTS

ACV Table 2 I/O Pins

INPUTS	#PINS
AC <3:0> H	4
B CLK L	1
BUS 4 H	1
CS PARITY ERROR H	1
D CLK ENABLE H	1
D SIZE <1:0> H	2
FP RES OP L	1
LATCHED BUS <3:0> H	4
LATCHED WCTRL <5:0> H	6
M CLK ENABLE H	1
MAD <02:00> H	3
PAGE BNDRY H	1
PHASE 1 H	1
PREFETCH L	1
TB VALID H	1
UB ADDRESS H	1
UTRAP L	1
WBUS <27:24> H	4
OUTPUTS (TOTEM POLE UNLESS OTHERWISE NOTED)	#PINS
ACV H	1
ENC UTRAP <2:0> L	3
FORCE MA 09 H	1
MICRO VECTOR <1:0> H	2(TRI-STATE)
PROC INIT H	1
PTE CHECK OR PROBE H	1

2.1 The following is a list of the code assignments for the BUS FUNCTION Micro field:

0. READ PHYSICAL ADDRESS
1. PROCESSOR INITIALIZE
2. READ, NO MICRO-TRAP
3. I/O INITIALIZE
4. READ LOCK TIMEOUT TEST
5. NOP
6. READ, SECOND REFERENCE
7. (RESERVED)
8. WRITE PHYSICAL ADDRESS
9. REI CHECK
A. WRITE, SECOND REFERENCE
B. WRITE UNLOCK, SECOND REFERENCE
C. WRITE, NO MICRO-TRAP
D. (RESERVED)
E. WRITE LONGWORD, NO MICRO-TRAP
F. BUS GRANT
10. READ
11. READ LONGWORD
12. PTE ACCESS CHECK, WRITE
13. READ LOCK

14. READ WITH MODIFY INTENT
15. READ LONGWORD WITH MODIFY INTENT
16. PTE ACCESS CHECK, READ
17. PTE ACCESS CHECK, READ, KERNEL MODE
18. WRITE
19. WRITE LONGWORD
1A. WRITE IF NOT RMODE
1B. WRITE UNLOCK
1C. PROBE ACCESS, WRITE, MODE SPECIFIED
1D. PROBE ACCESS, WRITE
1E. PROBE ACCESS, READ, MODE SPECIFIED
1F. PROBE ACCESS, READ

Bus Functions are decoded from "LATCHED BUS 4 H"(FLIP-FLOP) and "LATCHED BUS <3:0> H"

2.2 The following is a partial list of code assignments for the WCTRL Micro field:

00. PSL <- (WBUS)
20. VA <- PC + ISIZE + (WBUS)
PC <- PC + ISIZE
21. VA <- VA SAVE + (WBUS)
22. VA <- VA + 4
23. MDR <- (WBUS)
24. PC <- (WBUS)
25. VA <- (WBUS)
26. MBUS <- WDR
27. MDR <- 0
28. TB DATA <- (WBUS)
29. TB VALID BIT <- 0
VA <- (WBUS)
2A. WDR <- (WBUS) UNROTATED
2B. MDR <- IR, ZERO EXTENDED
2C. PC <- PC + (WBUS)
2D. CACHE VALID BIT <- 0
VA <- (WBUS)
2E. WDR <- (WBUS)
2F. MDR <- OSR, ZERO EXTENDED
30. STATUS/CONTROL REGISTER <- WBUS<27:24>
31. PREVIOUS MODE REGISTER <- WBUS<23:22>
32. WBUS<27:24> <- STATUS/CONTROL REGISTER
33. BUS GRANT
34. STATUS/CONTROL ADDRESS REGISTER <- WBUS<27:24>
35. IS/CURRENT MODE REGISTER <- WBUS<26:24>
37. REI CHECK
38. ASTLVL REGISTER <- WBUS<26:24>
39. (RESERVED)
3A. WBUS<26:24> <- ASTLVL REGISTER
3B. (RESERVED)
3C. HIGHEST SOFTWARE IPR REGISTER <- WBUS<20:16>
3D. IPL REGISTER <- WBUS<20:16>
3E. NOP
3F. WBUS<20:16> <- IPL OF HIGHEST IPR

WCTRL Functions are decoded from "LATCHED WCTRL <5:0> H"

2.3 Flip-Flop Definitions:

FLIP-FLOPS CLOCKED ON THE FALLING EDGE OF "B CLK L":

#"PROC INIT H" - JK FLIP-FLOP
J INPUT:

"PHASE 1 H" & (Specified Bus Function is PROCESSOR INITIALIZE)

K INPUT:

"PHASE 1 H" & (Specified Bus Function is NOT PROCESSOR INITIALIZE)

#"BOUNDARY H" - D FLIP-FLOP
D INPUT:

"PAGE BNDRY H"

FLIP-FLOPS CLOCKED ON THE RISING EDGE OF "B CLK L":

#"LATCHED BUS 4 H" - JK FLIP-FLOP
J INPUT:

"BUS 4 H" & "M CLK ENABLE H"

K INPUT:

"BUS 4 L" & "M CLK ENABLE H"

#"MME ADD H" - ENABLED D FLIP-FLOP CLOCK ENABLE:

"D CLK ENABLE H" &
[STATUS/CONTROL ADDRESS REGISTER <- WBUS <27:24> (WCTRL Decode)

D INPUT:

"WBUS 27 L" & "WBUS 26 L" & "WBUS 25 L" & "WBUS 24 L"

The flop is DC CLEARED when "PROC INIT H" is TRUE.

#"MME H" - ENABLED D FLIP-FLOP CLOCK ENABLE:

"D CLK ENABLE H" & "MME ADD H" &
[STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)
D INPUT:

"WBUS 24 H"

The flop is DC CLEARED when "PROC INIT H" is TRUE.

2.4 Current Mode Register: The CURRENT MODE REGISTER is a 2 bit edge triggered register which is clocked on the RISING edge of "B CLK L" if:

"D CLK ENABLE H" &
[IS/CURRENT MODE REGISTER <- WBUS <26:24> (WCTRL Decode) +
PSL <- (WBUS) (WCTRL Decode)

REGISTER Inputs are: "WBUS <25:24> H"
REGISTER Outputs are: "CUR MODE <1:0> H"

2.5 Micro-Trap Conditions:

CONTROL STORE PARITY ERROR Micro-Trap exists if:

"CS PARITY ERROR H"

FPA RESERVED OPERAND Micro-Trap exists if:

"FP RES OP H"

UNALIGNED DATA Micro-Trap exists if the specified Bus Function is one of the following:

WRITE
WRITE UNLOCK
WRITE IF NOT RMODE
READ WITH MODIFY INTENT
READ
READ LOCK
PROBE ACCESS, READ
PROBE ACCESS, READ, MODE SPECIFIED
PROBE ACCESS, WRITE
PROBE ACCESS, WRITE, MODE SPECIFIED
PTE ACCESS CHECK, READ
PTE ACCESS CHECK, WRITE
PTE ACCESS CHECK, READ, KERNEL MODE

AND the combination of "D SIZE <1:0> H" and "MAD <01:00> H" is marked "UNAL" in the following Table:

	MAD<01:00>			
D-SIZE<1:0>	00	01	10	11
00				
01				UNAL
10		UNAL	UNAL	UNAL
11		UNAL	UNAL	UNAL

UNALIGNED DATA, WRITE UNLOCK Micro-Trap exists if UNALIGNED DATA Micro-Trap exists AND the specified Bus Function is WRITE UNLOCK.

UNALIGNED UNIBUS DATA Micro-Trap exists if the specified Bus Function is not BUS GRANT and "UB ADDRESS H" is TRUE and "CROSS PAGE H" is FALSE and:

"PREPETCH H" is TRUE or The specified Bus Function is neither an ACCESS PROBE nor a PTE ACCESS CHECK, and the combination of "D SIZE <1:0> H" and "MAD <01:00> H" is marked "UNAL" in the following Table:

	MAD<01:00>			
D-SIZE<1:0>	00	01	10	11
00				
01		UNAL		UNAL
10	UNAL	UNAL	UNAL	UNAL
11	UNAL	UNAL	UNAL	UNAL

WRITE CROSSING PAGE BOUNDARY Micro-Trap exists if "CROSS PAGE H" is TRUE and the specified Bus Function is one of the following:

WRITE
WRITE IF NOT RMODE
PROBE ACCESS, READ
PROBE ACCESS, READ, MODE SPECIFIED
PROBE ACCESS, WRITE
PROBE ACCESS, WRITE, MODE SPECIFIED

"CROSS PAGE H" is TRUE if the specified BUS FUNCTION is one of the following:

WRITE
WRITE UNLOCK
WRITE IF NOT RMODE
PROBE ACCESS, READ
PROBE ACCESS, READ, MODE SPECIFIED
PROBE ACCESS, WRITE
PROBE ACCESS, WRITE, MODE SPECIFIED

and "MME H" is TRUE and "BOUNDARY H" is TRUE and "PREFETCH H" is FALSE and "FP RES OP H" is FALSE and the combination of "D SIZE <1:0> H" and "MAD <02:00> H"is marked "EOP" in the following Table:

		MAD<02:00>							
		000	001	010	011	100	101	110	111
D SIZE <1:0>	00								
	01								EOP
	10						EOP	EOP	EOP
	11		EOP	EOP	EOP	EOP	EOP	EOP	EOP

WRITE UNLOCK CROSSING PAGE BOUNDARY Micro-Trap exists if "CROSS PAGE H" is TRUE and the specified Bus Function is one of the following:

WRITE UNLOCK.
PROBE ACCESS, READ
PROBE ACCESS, READ, MODE SPECIFIED
PROBE ACCESS, WRITE
PROBE ACCESS, WRITE, MODE SPECIFIED

CHIP OUTPUTS

>>> "ACV H" is TRUE (HIGH) if the intended access violates the access protection code as defined per ACV Table 3.

If "PREFETCH H" is TRUE, the intended access is READ, and the MODE (CM) bits are: "CUR MODE <1:0> H".

If "PREFETCH H" is FALSE, and the specified Bus Function is one of the following, no access check is made, and "ACV H" is FALSE:

READ, NO MICRO-TRAP
WRITE, NO MICRO-TRAP
WRIrE LONGWORD, NO MICRO-TRAP
BUS GRANT
IO INITIALIZE

Of the remaining Bus Functions, the following are checked for READ access:

READ
READ, SECOND REFERENCE
READ LONGWORD
READ PHYSICAL ADDRESS
PROBE ACCESS, READ
PROBE ACCESS, READ, MODE SPECIFIED
PTE ACCESS CHECK, READ
PTE ACCERSS CHECK, READ, KERNEL MODE
PROCESSOR INITIALIZE

All other Bus Functions are checked for WRITE access.

If "PREFETCH H" is FALSE and the specified Bus Function is:

PTE ACCESS CHECK, READ, KERNEL MODE

the MODE (CM) bits are forced to KERNEL MODE (00).

If "PREFETCH H" is FALSE and the specified Bus Function is:

ACCESS PROBE, READ, MODE SPECIFIED
or
ACCESS PROBE, WRITE, MODE SPECIFIED

the MODE (CM) bits are: "WBUS <25:24> H".

Otherwise, the MODE (CM) bits are: "CUR MODE <1:0> H".

>>> "ENC UTRAP <2:0> L"

The following MICRO-TRAP conditions are detected and prioritized:

(HIGHEST) 1. CONTROL STORE PARITY ERROR
2. FPA RESERVED OPERAND
3. UNALIGNED UNIBUS DATA
4. WRITE CROSSING PATE BOUNDARY
5. WRITE UNLOCK CROSSING PAGE BOUNDARY
6. UNALIGNED DATA, WRITE UNLOCK
7. UNALIGNED DATA

If one or more of these MICRO-TRAPS exists, "ENC UTRAP <2:0> L" reflect the identity of the highest priority MICRO-TRAP as shown in the following table:

MICRO-	"ENC UTRAP		
TRAP #	2 L"	1 L"	0 L"
1	L	L	L
2	L	L	H
3	L	H	H
4	H	L	H
5	H	L	L
6	L	H	L
7	H	H	L

If none of these MICRO-TRAPS exists, "ENC UTRAP <2:1> L" are HIGH.

"ENC UTRAP 0 L" is also HIGH unless "UTRAP L" is HIGH and:

The specified Bus Function is a PTE ACCESS CHECK

or

"CROSS PAGE H" is TRUE and the specified Bus Function is an ACCESS PROBE.

>>> "FORCE MA 09 H" is TRUE if:

"PHASE 1 L" & "INVAL LATCH H"

"INVAL LATCH H" is the HIGH TRUE output of a latch which is enabled while "PHASE 1 H" is TRUE.

Latch input is: [29 (WCTRL Decode)]
>>> "MICRO VECTOR <1:0> H" are enabled if "UTRAP H" is FALSE and the specified Bus Function is a PTE ACCESS CHECK or an ACCESS PROBE.

When enabled, "MICRO VECTOR 1 H" is TRUE (HIGH) if:

(ACCESS PROBE) & ("TB VALID H" + "MME L") +

"TB VALID H" & ["ACV H" (Chip Output) is FALSE]

When enabled, "MICRO VECTOR 0 H" is TRUE (HIGH) if:

(ACCESS PROBE) & "MME L" +

["ACV H" (Chip Output) is FALSE] &
["TB VALID H" + (PTE ACCESS CHECK)]

>>> "PROC INIT H" is TRUE (HIGH) if the "PROC INIT H" flip-flop is set.

>>> "PTE CHECK OR PROBE H" is TRUE (HIGH) if "UTRAP H" is FALSE, and the specified Bus Function is an ACCESS PROBE or a PTE ACCESS CHECK.

ACV Table 3

ACCESS CONTROL DATA

CODE		MNEMONIC					
DECIMAL	BINARY		K	E	S	U	COMMENT
0	0000	NA	-	-	-	-	no ACCESS
1	0001			UNPREDICTABLE			RESERVED
2	0010	KW	RW	-	-	-	
3	0011	KR	R	-	-	-	
4	0100	UW	RW	RW	RW	RW	ALL ACCESS
5	0101	EW	RW	RW	-	-	
6	0110	ERKW	RW	R	-	-	
7	0111	ER	R	R	-	-	
8	1000	SW	RW	RW	RW	-	
9	1001	SREW	RW	RW	R	-	
10	1010	SRKW	RW	R	R	-	
11	1011	SR	R	R	R	-	
12	1100	URSW	RW	RW	RW	R	
13	1101	UREW	RW	RW	R	R	
14	1110	URKW	RW	R	R	R	
15	1111	UR	R	R	R	R	

-	-	no access	K - Kernel
R	-	read only	E - Executive
RW	-	read write	S - Supervisor
			U - User

The access is allowed if:

```
        (CODE NEWU 0) AND
            ((CODE EQLU4) or (CM LSSU WM) OR (READ AND (CM LEQU RM)))

        CM is current mode
        RM is left 2 bits of code
        WM is one's complement of right 2 bits of code
```

```
                          -----------------
             BUS 4 H --->!01            48!<--- D SIZE 0 H
     M CLK ENABLE H --->!02            47!<--- B CLK L
          WBUS 24 H --->!03            46!---> PROC INIT H
             AC 0 H --->!04            45!<--- PAGE BNDRY H
             AC 3 H --->!05            44!<--- WBUS 25 H
             AC 1 H --->!06            43!<--- WBUS 27 H
             AC 2 H --->!07            42!<--- WBUS 26 H
            UTRAP L --->!08            41!<--- PHASE 1 H
              ACV H --->!09            40!---> FORCE MA 09 H
   LATCHED BUS 0 H --->!10 .......... 39!<--- LATCHED WCTRL 0 H
   LATCHED BUS 3 H --->!11 .        . 38!---- GROUND
                VGA ----!12 . LID    . 37!<--- LATCHED BUS 1 H
                VCC ----!13 . DOWN   . 36!<--- LATCHED WCTRL 3 H
         TB VALID H --->!14 .        . 35!---- GROUND
PTE CHECK OR PROBE H <---!15 .......... 34!<--- MA 00 H
   MICRO VECTOR 0 H <---!16            33!<--- LATCHED WCTRL 1 H
   MICRO VECTOR 1 H <---!17            32!<--- MA 01 H
       ENC UTRAP 1 L <---!18            31!<--- LATCHED WCTRL 5 H
       ENC UTRAP 0 L <---!19            30!<--- D CLK ENABLE H
        UB ADDRESS H --->!20            29!<--- MA 02 H
       ENC UTRAP 2 L <---!21            28!<--- LATCHED WCTRL 2 H
         PREFETCH L --->!22            27!<--- LATCHED WCTRL 4 H
  CS PARITY ERROR H --->!23            26!<--- D SIZE 1 H
   LATCHED BUS 2 H --->!24            25!<--- FP RES OP L
                          -----------------
```

FIGURE

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.2 ADD ADDRESS CHIP (DC609)

1. GENERAL DESCRIPTION:

The ADDRESS CHIP is an 8 bit slice designed to accomplish address manipulation on one byte of the memory address. Data is input to the address chips from the WB and driven from the address chips on the MAD.

ADD Table 1

Pin Identification and Gate Type

PIN #	PIN ID	GATE TYPE
1	ENA VA SAVE L	GA1TNG
2	PAGE BNDRY H	GA1TPN
3	BSRC SEL S0 H	GA1TNG
5	WB 06 H	GA1TNG
6	WB 05 H	GA1TNG
7	WB 07 H	GA1TNG
8	ASRC SEL S0 H	GA1TNG
9	CARRY GEN 1 H	GA1TTN
10	CARRY GEN 2 H	GA1TTN
11	BSRC SEL S1 H	GA1TNG
14	CARRY PROP H	GA1TTN
15	WB 04 H	GA1TNG
16	ASRC SEL S2 H	GA1TNG
17	WB 00 H	GA1TNG
18	WB 03 H	GA1TNG
19	WB 02 H	GA1TNG
20	ASRC SEL S1 H	GA1TNG
21	WB 01 H	GA1TNG
22	CARRY IN L	GA1TNG
23	MSEQ VA 00 H	GA1TTG
24	ADD CHIP ID H	GA1TNG
25	XB PC 00 H	GA1TTG
26	FORCE MA 01 H	CA1TNI
27	MAD 01 H	GA1TTG
28	MAD 02 H	GA1TTG
29	INC CARRY IN L	GA1TNG
30	XB PC 01 H	GA1TTG
31	MAD 00 H	GA1TTG
32	XB PC 02 H	GA1TTG
33	MAD 03 H	GA1TTG
34	LATCH MA L	GA1TNG
36	ENA PC BACKUP L	GA1TNG
37	B CLK L	GA1TNG
39	MA SELECT S0 H	GA1TNG
40	COMP MODE H	GA1TNG
41	MA SELECT S1 H	GA1TNG
42	ENA PC L	GA1TNG
43	ENA VA L	GA1TNG
44	MAD 04 H	GA1TTG
45	MAD 07 H	GA1TTG
46	INC CARRY OUT L	GA1TTN
47	MAD 05 H	GA1TTG
48	MAD 06 H	GA1TTG

2. Performance Requirements:

The ADDRESS CHIP contains four registers:

1. VA (VIRTUAL ADDRESS REGISTER)
2. PC (PROGRAM COUNTER)
3. VA SAVE
4. PC BACKUP

VA and VA SAVE are loaded directly from the output of the adder. PC and PC BACKUP are loaded from the output of the VA SAVE register. Since the VA SAVE register is a transparent latch, PC and PC BACKUP may also be loaded with information from the adder output by enabling the VA SAVE register while the PC or PC BACKUP is being enabled.

Sources for the A input to the adder are:

1. 0
2. +1
3. +2
4. +4
5. WB

Sources for the B input to the adder are:

1. 0
2. VA
3. VA SAVE
4. PC

2.1 The MA is a transparent latch. While enabled, it passes information from the MA multiplexer directly to the MAD. When the enable goes false, the MA retains the information that is present on the inputs at that time until the enable becomes true again.

Sources for the MA are:

1. VA
2. PC
3. PC BACKUP
4. PC INCREMENTER

The PC INCREMENTER continuously provides PC+4 for pre-fetching.

2.2 CHIP I/O SUMMARY (See "FUNCTIONS" for Pin Functions): See ADD Figure 1 for Pin Identification.

POWER/GROUND	4 pins
WB	8 pins
MAD	8 pins
BCLK	1 pin
FORCE MA 01	1 pin
ADD CHIP ID	1 pin
PAGE BNDRY	1 pin
MSEQ VA00	1 pin
XB PC<02:00>	3 pins
ENA VA L	1 pin
ENA VA SAVE	1 pin
ENA PC	1 pin
ENA PC BACKUP	1 pin
LATCH MA	1 pin
ASRC SEL	3 pins
BSRC SEL	2 pins
MA SELECT	2 pins
CARRY IN	1 pin
CARRY GEN	2 pins
CARRY PROP	1 pin
INC CARRY IN	1 pin
INC CARRY OUT	1 pin
COMP MODE	1 pin

2.3 FUNCTIONS

WB: The WB may be sourced onto the A INPUT of the adder.

MAD: The MAD is driven from the MA latch. If the latch is enabled, the MAD contains information selected by the MA MULTIPLEXER. If the MA latch is not enabled, the MAD contains the information that was present on the latch inputs at the time the enable went false.

BCLK: For simulation purposes, the VA, PC and PC BACKUP may be considered as edge triggered registers which change state on the low to high transition of BCLK. VA SAVE is a transparent latch which is loaded during BCLK low time if ENA VA L is true. While the latch is being loaded, the outputs follow the inputs.

FORCE MA 01: When high, overrides normal MA functions and forces bit 01 of the MAD to a 1.

ADD CHIP ID: The INCREMENTER, PAGE BNDRY and CONSTANT functions must be performed differently depending on whether a particular ADDRESS CHIP is connected as the least significant byte or one of the three most significant bytes in the address word. When connected as the least significant byte:

1. The INCREMENTER adds a one to the PC in bit position 02 and INC CARRY OUT is true if bits PC<07:02> are all ones.

2. PAGE BNDRY is true if bits VA<07:03> are all ones.

3. The CONSTANTS (+1,+2,+4) are enabled when selected on the A INPUT MULTIPLEXER.

When connected as one of the three most significant bytes:

1. The INCREMENTER adds a one to the PC in bit position 00 only when INC CARRY IN is true. INC CARRY OUT is true when bits PC<07:00> are all ones and INC CARRY IN is true.

2. PAGE BNDRY is true if bit VA<00> is a one.

3. The CONSTANTS are continuously disabled.

When ADD CHIP ID is low, the ADDRESS CHIP functions as the least significant byte. When ADD CHIP ID is high, the ADDRESS CHIP functions as one of the three most significant bytes.

PAGE BNDRY: If ADD CHIP ID is low, PAGE BNDRY is high when bits VA<07:03> are all ones. If ADD CHIP ID is high, PAGE BNDRY is high when VA<00> is a one.

MSEQ VA 00: MSEQ VA 00 is high when bit VA00 is a one.

XB PC<02:00>: XB PC<02:00> are high when bits PC<02:00> respectively are ones.

ENA VA L (LOW TRUE): If ENA VA L is true during BCLK low time, the information on the output of the adder will be loaded into the VA register at the low to high transition of BCLK.

ENA VA SAVE (LOW TRUE): When ENA VA SAVE is true, the VA SAVE latch will be enabled during BCLK low time. While the VA SAVE latch is enabled, the outputs will follow the inputs. At the low to high transition of BCLK, the information present on the inputs will be retained.

ENA PC (LOW TRUE): If ENA PC is true during BCLK low time, the information at the output of the VA SAVE latch will be loaded into the PC register on the low to high transition of BCLK. If ENA VA SAVE is also true, the VA SAVE latch will be transparent, and the information loaded into PC will be the adder output.

ENA PC BACKUP (LOW TRUE): If ENA PC BACKUP is true during BCLK low time, the information at the output of the VA SAVE latch will be loaded into the PC BACKUP register on the low to high transition of BCLK. If ENA VA SAVE is also true, the VA SAVE latch will be transparent, and the information loaded into PC BACKUP will be the adder output.

LATCH MA: While LATCH MA is high, the MA latch is transparent and passes data directly from the MA MULTIPLEXER to the MAD. When LATCH MA is low at a low to high transition of BCLK, the data that was present on the latch inputs at the previous high to low transiton of BCLK is retained in the latch until the next low to high transition of BCLK occurs with LATCH MA high, at which time the latch becomes transparent again.

ASRC SEL: The ASRC SEL pins control what data will be enabled onto the A INPUT of the adder according to the following Table:

ASRC SEL S2 H	ASRC SEL S1 H	ASRC SEL S0 H	A INPUT
LOW	LOW	LOW	0
LOW	LOW	HIGH	+1
LOW	HIGH	LOW	+2
LOW	HIGH	HIGH	+4
HIGH	X	X	WB

BSRC SEL: The BSRC SEL pins control what data will be enabled onto the B INPUT of the adder according to the following Table:

BSRC SEL S1 H	BSRC SEL S0 H	B INPUT
LOW	LOW	0
LOW	HIGH	PC
HIGH	LOW	VA SAVE
HIGH	HIGH	VA

MA SELECT: The MA SELECT pins control what data will be enabled onto the inputs of the MA latch according to the following Table:

MA SELECT S1 H	MA SELECT S0 H	MA INPUT
LOW	LOW	INCREMENTER
LOW	HIGH	PC BACKUP
HIGH	LOW	PC
HIGH	HIGH	VA

CARRY IN: CARRY IN is the carry into the adder from a less significant stage. CARRY IN is low for a carry and high for no carry. It will be tied high on the least significant chip.

CARRY GEN: CARRY GEN and CARRY PROP are the adder carry look-ahead signals, and are designed to be compatable with the 74S182 carry look-ahead generator. CARRY GEN is high when the sum of the A and B adder inputs (in one chip) is greater than or equal to FF (HEX). Two output pins drive this signal to provide the 16ma required by the G input on a 74S182.

CARRY PROP: (See CARRY GEN above). CARRY PROP is high when both the A and B adder inputs at any bit position (within the chip) are ones.

INC CARRY IN: INC CARRY IN is the carry into the INCREMENTER from a less significant stage; low = carry, high = no carry. If ADD CHIP ID is low, INC CARRY IN has no effect. If ADD CHIP ID is high and INC CARRY IN is low, the INCREMENTER adds one to the value of PC in that chip.

INC CARRY OUT: INC CARRY OUT is low for an INCREMENTER carry out and high for no carry. The INCREMENTER in a chip generates a carry (low) if ADD CHIP ID is low and bits PC<07:02> are all ones or if ADD CHIP ID is high, bits PC<07:00> are all ones, and INC CARRY IN is low.

COMP MODE: If COMP MODE is high, the outputs of the MA MULTIPLEXER are forced to zeroes.

```
                          ---------------
ENA VA SAVE L --->!01              48!---> MAD 06 H
  PAGE BNDRY H <---!02              47!---> MAD 05 H
BSRC SEL SO H --->!03              46!---> INC CARRY OUT L
                  !04              45!---> MAD 07 H
      WB 06 H --->!05              44!---> MAD 04 H
      WB 05 H --->!06              43!<--- ENA VA L
      WB 07 H --->!07              42!<--- ENA PC L
ASRC SEL S0 H --->!08              41!<--- MA SELECT S1 H
CARRY GEN 1 H <---!09              40!<--- COMP MODE H
CARRY GEN 2 H <---!10 ....... 39!<--- MA SELECT S0 A
BSRC SEL S1 H --->!11 .       . 38!---- GROUND
          VGA ----!12 . LID . 37!<--- B CLK L
          VCC ----!13 . DOWN. 36!<--- ENA PC BACKUP L
 CARRY PROP H <---!14 .       . 35!---- GROUND
      WB 04 H --->!15 ....... 34!<--- LATCH MA L
ASRC SEL S2 H --->!16              33!---> MAD 03 H
      WB 00 H --->!17              32!---> XB PC 02 H
      WB 03 H --->!18              31!---> MAD 00 H
      WB 02 H --->!19              30!---> XB PC 01 H
ASRC SEL S1 H --->!20              29!<--- INC CARRY IN L
      WB 01 H --->!21              28!---> MAD 02 H
   CARRY IN L --->!22              27!---> MAD 01 H
 MSEQ VA 00 H <---!23              26!<--- FORCE MA 01 H
ADD CHIP ID H --->!24              25!---> XB PC 00 H
                          ---------------
```

ADD FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.3 ADDRESS CONTROL CHIP (ADK-DC626)

1. GENERAL DESCRIPTION:

The ADK chip generates most of the signals necessary to control the translation buffer and contains the status/control registers associated with the translation buffer. In addition, ADK generates several control signals for the MDR and ADD bit slice chips.

ADK TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	AMUX SEL S1 H		GA1TTN
2	DBUS SEL S1 H		GA1TTN
3	WBUS 26 H		GA1TZF
4	WBUS 25 H		GA1TZF
5	WBUS 24 H		GA1TZF
6	WBUS 27 H		GA1TZF
7	TB GRP Ø WR H		GA1TTN
8	TB HIT Ø H	I_{OL} = 2ØmA	GA1TCF
9	TB GRP 1 WR H		GA1TTN
1Ø	TB HIT 1 H	I_{OL} = 2ØmA	GA1TCF
11	ENA VA L		GA1TTN
14	LATCHED WCTRL 2 H		GA1TNF
15	BSRC SEL SØ H		GA1TTN
16	LATCHED WCTRL 5 H		GA1TNF
17	LATCHED WCTRL Ø H		GA1TNF
18	LATCHED WCTRL 3 H		GA1TNF
19	BSRC SEL S1 H		GA1TTN
2Ø	LATCHED WCTRL 1 H		GA1TNF
21	LATCHED WCTRL 4 H		GA1TNF
22	CLK SEL SØ H		GA1TTN
23	CLK SEL S1 H		GA1TTN
24	PHASE 1 H		GA1TNF
25	DBUS SEL SØ H		GA1TTN
26	TB OUTPUT ENA L	I_{OL} = 12 mA	GA1TTN
27	LATCHED BUS 2 H		GA1TNF
28	COMP MODE H		GA1TTN
29	PREFETCH L		GA1TNF
3Ø	LATCHED BUS 1 H		GA1TNF
31	PTE CHECK L		GA1TTN
32	LATCHED BUS Ø H		GA1TNF
33	LATCHED BUS 3 H		GA1TNF
34	TB PARITY ENA H		GA1TTN
36	BUS 4 H		GA1TNF
37	D CLK ENABLE H		GA1TNF
39	M CLK ENABLE H		GA1TNF
4Ø	STATUS VALID L		GA1TNF
41	RTUT DINH L		GA1TNF
42	PSL CM H		GA1TNF
43	SNAPSHOT CMI L		GA1TNF
44	WRITE VECT OCC L		GA1TNF
45	MMUX SEL S1 H		GA1TNH
46	DST RMODE H		GA1TNF
47	AMUX SEL SØ H		GA1TTN
48	B CLK L		GA1TNF

2 **Performance:**

ADK I/O PINS

INPUTS	#PINS
B CLK L	1
BUS 4 H	1
D CLK ENABLE H	1
DST RMODE H	1
LATCHED BUS <3:0> H	4
LATCHED WCTRL <5:0> H	6
M CLK ENABLE H	1
MMUX SEL S1 H	1
PHASE 1 H	1
PREFETCH L	1
PSL CM H	1
SNAPSHOT CMI L	1
STATUS VALID L	1
RTUT DINH L	1
WRITE VECT OCC L	1

BI-DIRECTIONAL LINES (TRI-STATE UNLESS OTHERWISE NOTED)

TB HIT <1:0> H	2(OPEN-COLL.)
WBUS <27:24> H	4

OUTPUTS (TOTEM POLE UNLESS OTHERWISE NOTED)

AMUX SEL S<1:0> H	2
BSRC SEL S<1:0> H	2
CLK SEL S<1:0> H	2
COMP MODE H	1
DBUS SEL S<1:0> H	2
ENA VA L	1
PTE CHECK L	1
TB GRP <1:0> WR H	2
TB OUTPUT ENA L	1
TB PARITY ENA H	1

2.1 **Code Assignments:** The following is a list of the code assignments for the BUS FUNCTION Micro field:

0. READ PHYSICAL ADDRESS
1. PROCESSOR INITIALIZE
2. READ, NO MICRO-TRAP
3. I/O INITIALIZE
4. READ LOCK TIMEOUT TEST
5. NOP
6. READ, SECOND REFERENCE
7. NOP
8. WRITE PHYSICAL ADDRESS
9. REI CHECK
A. WRITE, SECOND REFERENCE
B. WRITE UNLOCK, SECOND REFERENCE
C. WRITE, NO MICRO-TRAP
D. NOP
E. WRITE LONGWORD, NO MICRO-TRAP
F. BUS GRANT
10. READ
11. READ LONGWORD
12. PTE ACCESS CHECK, WRITE
13. READ LOCK
14. READ WITH MODIFY INTENT
15. READ LONGWORD WITH MODIFY INTENT
16. PTE ACCESS CHECK, READ
17. PTE ACCESS CHECK, READ, KERNEL MODE
18. WRITE
19. WRITE LONGWORD
1A. WRITE IF NOT RMODE
1B. WRITE UNLOCK
1C. PROBE ACCESS, WRITE, MODE SPECIFIED
1D. PROBE ACCESS, WRITE
1E. PROBE ACCESS, READ, MODE SPECIFIED
1F. PROBE ACCESS, READ

Bus Functions are decoded from "LATCHED BUS 4 H"(FLIP-FLOP) and "LATCHED BUS <3:0> H".

2.2 **Code Assignments (Partial List):** The following is a partial list of code assignments for the WCTRL Micro field:

00.	PSL <- (WBUS)

20.	VA <- PC + ISIZE + (WBUS)
	PC <- PC + ISIZE

21.	VA <- VA SAVE + (WBUS)
22.	VA <- VA + 4
23.	MDR <- (WBUS)
24.	PC <- (WBUS)
25.	VA <- (WBUS)
26.	MBUS <- WDR
27.	MDR <- 0
28.	TB DATA <- (WBUS)
29.	TB VALID BIT <- 0 VA <- (WBUS)
2A.	WDR <- (WBUS) UNROTATED
2B.	MDR <- IR, ZERO EXTENDED
2C.	PC <- PC + (WBUS)
2D.	CACHE VALID BIT <- 0 VA <- (WBUS)
2E.	WDR <- (WBUS)
2F.	MDR <- OSR, ZERO EXTENDED
30.	STATUS/CONTROL REGISTER <- WBUS<27:24>
31.	PREVIOUS MODE REGISTER <- WBUS<23:22>
32.	WBUS<27:24> <- STATUS/CONTROL REGISTER
33.	BUS GRANT
34.	STATUS/CONTROL ADDRESS REGISTER <- WBUS<27:24>
35.	IS/CURRENT MODE REGISTER <- WBUS<26:24>
37.	REI CHECK
38.	ASTLVL REGISTER <- WBUS<26:24>
39.	(RESERVED)
3A.	WBUS<26:24> <- ASTLVL REGISTER
3B.	(RESERVED)
3C.	HIGHEST SOFTWARE IPR REGISTER <- WBUS<20:16>
3D.	IPL REGISTER <- WBUS<20:16>
3E.	NOP
3F.	WBUS<20:16> <- IPL OF HIGHEST IPR

WCTRL Functions are decoded from "LATCHED WCTRL <5:0> H".

2.3 **Internal Signal Definitions:**

#"STATUS VAL H" - D FLIP-FLOP (RISING EDGE OF "B CLK L")
D INPUT:

"STATUS VALID H"

The flop is DC CLEARED when "STATUS VALID H" is FALSE.

#"LATCHED BUS 4 H" - JK FLIP-FLOP
J INPUT:

"M CLK ENABLE H" & "BUS 4 H"

K INPUT:

"M CLK ENABLE H" & "BUS 4 L"

#"PREFETCH DEL H - D FLIP-FLOP
D INPUT:

"PREFETCH H"

The flop is DC cleared if:

"PREFETCH L" + "PROC INIT H"

#"PREFETCH CYC H - D FLIP-FLOP
D INPUT:

"PREFETCH DEL H"

The flop is DC PRESET when "PREFETCH DEL H" is TRUE.

*"BUS CYC DEC H" is TRUE for any of the following Bus Function Decodes:

READ PHYSICAL ADDRESS
RE\D, NO MICRO-TRAP
READ LOCK TIMEOUT TEST
READ, SECOND REFERENCE
WRITE PHYSICAL ADDRESS
WRITE, SECOND REFERENCE
WRITE UNLOCK, SECOND REFERENCE
WRITE, NO MICRO-TRAP
WRITE LONGWORD, NO MICRO-TRAP
BUS GRANT
READ
READ LONGWORD
READ LOCK
READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
WRITE
WRITE LONGWORD
WRITE UNLOCK

"DST RMODE L" & WRITE IF NOT RMODE

*"BUS GRANT DEC H" is TRUE for: (Bus Function Decode)

BUS GRANT

*"PHYS DEC H" is TRUE for: (Bus Function Decodes)

READ PHYSICAL ADDRESS
WRITE PHYSICAL ADDRESS

*"PROC INIT H" is TRUE if:

"PHASE 1 L" & PROCESSOR INITIALIZE(Bus Function Decode)

*"WRITE TB H" is TRUE if:

[28 + 29](WCTRL Decodes)

*"CA INVAL H" is TRUE if:

[2D](WCTRL Decode)

*"PTE CHECK H" is TRUE for: (Bus Function Decodes)

PTE ACCESS CHECK, WRITE
PTE ACCESS CHECK, READ
PTE ACCESS CHECK, READ, KERNEL MODE

*"BUS REQ H" is TRUE if:

"BUS CYC DEC H" & "PREFETCH L"

*"DEST VA H" is TRUE if:

[20 + 21 + 22 + 25 + 29 + 2D](WCTRL Decodes)

*"FULL ADD H" is TRUE for: (Bus Function Decodes)

READ PHYSICAL ADDRESS
READ, NO MICRO-TRAP
WRITE PHYSICAL ADDRESS
WRITE, NO MICRO-TRAP
WRITE LONGWORD, NO MICRO-TRAP
PROBE ACCESS, WRITE, MODE SPECIFIED
PROBE ACCESS, WRITE
PROBE ACCESS, READ, MODE SPECIFIED
PROBE ACCESS, READ

*"LATCHED HIT 1 H" is the HIGH TRUE output of a latch which is enabled during:

"PHASE 1 H"

LATCH Input is:

"TB HIT 1 H"

*"LATCHED HIT 0 H" is the HIGH TRUE output of a latch which is enabled during:

"PHASE 1 H"

LATCH Input is:

"TB HIT 0 H"

>>NOTE: "ADD REG ENA H" has not yet been defined.
"INVAL CHECK H" has not yet been defined.

*"MEM REQ H" is TRUE if:

["PREFETCH DEL H" +
("BUS CYC DEC H" & "PREFETCH L" & "REPLACEMENT L")] &
("ADD REG ENA H" + "CYC IN PROG L")

*"REPLACEMENT H" is TRUE if:

"STATUS VAL H" & "ADD ENA DEL L" & "READ H"

#"ADD REG ENA H - JK FLIP-FLOP
J INPUT:

"MEM REQ H" & "INVAL CHECK L"

K INPUT:

"PREFETCH CYC H" + "M CLK ENABLE H"

#"ADD ENA DEL H - D FLIP-FLOP
D INPUT:

"ADD REG ENA H"

The flop is DC PRESET when "ADD REG ENA H" is TRUE.

*"RESET ADD ENA H" is TRUE if:

"M CLK ENABLE H" + "PREFETCH CYC H"

#"INVAL CHECK H - JK FLIP-FLOP
J INPUT:

("MMUX SEL S1 L" & "SNAPSHOT CMI H") &
[("ADD REG ENA H" & "RESET ADD ENA H") + "MEM REQ L"

K INPUT:

"SNAPSHOT CMI L"

STATUS/CONTROL ADDRESS REGISTER: The S/C ADDRESS REGISTER is a 4 bit latch which is enabled during:

"B CLK H" & D CLK ENABLE H" &
STATUS/CONTROL ADDRESS REGISTER <- WBUS <27:24>(WCTRL Decode)

LATCH Inputs are: "WBUS <27:24> H"
LATCH Outputs are: "SC add <3:0> H"

STATUS/CONTROL REGISTERS

MME REGISTER: The MME REGISTER is a one bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 L" & "SC ADD 2 L" & "SC ADD 1 L" & "SC ADD 0 L" &
"STATUS/CONTROL REGISTER <- WBUS <27:24>(WCTRL Decode)

LATCH Input is: "WBUS 24 H"
LATCH Output is: "MME H"

The latch is DC cleared when "PROC INIT H" is TRUE.

WRITE VECTOR OCCURRED REGISTER: The WRITE VECTOR OCCURRED REGISTER is a one bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 L" & "SC ADD 2 L" & "SC ADD 1 H" & "SC ADD 0 L" &
"STATUS/CONTROL REGISTER <- WBUS <27:24>(WCTRL Decode)

LATCH Input is: "WBUS 24 H"
LATCH Output is" "WR VECT H"

The latch is DC cleared by:

"PROC INIT H" +
("BUS GRANT DEC H" & "PHASE 1 H" & "B CLK H")

The latch is DC PRESET when "WRITE VECT OCC H" is TRUE.

TB GROUP DISABLE REGISTER: The TB GROUP DISABLE REGISTER is a 4 bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 L" & "SC ADD 2 L" & "SC ADD 1 H" & "SC ADD 0 H" &
"STATUS/CONTROL REGISTER <- WBUS <27:24>(WCTRL Decode)

LATCH Inputs are: "WBUS <27:24> H"
LATCH Outputs are: "TB CTL <3:0> H"

The latches are DC cleared when "PROC INIT H" is TRUE. CURRENT MODE REGISTER: The CURRENT MODE REGISTER is a 2 bit latch which is enabled during:

```
   ("B CLK H" & "D CLK ENABLE H") &
   [IS/CURRENT MODE REGISTER <- WBUS <26:24>(WCRTL
   Decode) + PSL <- (WBUS)(WCTRL Decode)]
```

LATCH Inputs are: "WBUS <25:24> H"
LATCH Outputs are: "CUR MODE <1:0> H"

*"REPLACE 0 H - D FLIP-FLOP
D INPUT:

```
   ("TB CTL 3 H" & "TB CTL 2 L") +
   ("TB CTL 3 L" & "REPLACE 0 L")
```

The flop is DC PRESET when "PROC INIT H" is TRUE.

*"CYC IN PROG H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

```
   "ADD REG ENA H"
```

and reset by:

```
   "STATUS VAL H" + "PROC INIT H"
```

*"READ H" is the HIGH TRUE output of a latch which is enabled during:

```
   "ADD REG ENA H"
```

LATCH Input is:

```
   "PREFETCH H" + "LATCHED BUS 3 L"
```

*"CACHE CYC H" is TRUE if:

```
   ("ADD REG ENA H" + "CYC IN PROG L") &
   ("PREFETCH DEL H" + "BUS REQ H") &
   ("ADD ENA DEL H" + "STATUS VAL L" + "READ L")
```

*"PHYS ADD H" is TRUE if:

```
   ("INVAL CHECK L" & "CACHE CYC H") &
   ["MME L" + READ PHYSICAL ADDRESS(Bus Function Decode)
   + WRITE PHYSICAL ADDRESS(Bus Function Decode)]
```

SAVED MODE REGISTER: The SAVED MODE REGISTER is a 4 bit latch. Bits <2:0> are enabled during:

["B CLK H" & "D CLK ENABLE H" & "SC ADD 3 L" & "SC ADD 2 L" & "SC ADD 1 L" & "SC ADD 0 H" & STATUS/CONTROL REGISTER <- WBUS <27:24>(WCTRL Decode)] + ("BUS CYC DEC H" & "BUS GRANT DEC L" & "D CLK ENABLE H" &"B CLK H" & "RTUT DINH L")

Bit <3> is enabled during:

["B CLK H" & "D CLK ENABLE H" & "SC ADD 3 L" & "SC ADD 2 L" & "SC ADD 1 L" & "SC ADD 0 H" & STATUS/CONTROL REGISTER <- WBUS <27:24>(WCTRL Decode)] +

("ADD REG ENA H" & "READ H" & "LATCHED BUS 4 H" & "PREFETCH L" & "B CLK H")

LATCH Outputs are: "SAVED MODE <3:0> H"

"SAVED MODE 3 H" LATCH Input is:

("WBUS 27 H" & "BUS CYC DEC L") +
("LATCHED BUS 2 L" & "LATCHED BUS 1 L" & "BUS CYC DEC H")

"SAVED MODE 2 H" LATCH Input is:

("WBUS 26 H" & "BUS CYC DEC L") +
("PHYS ADD H" & "BUS CYC DEC H")

"SAVED MODE 1 H" LATCH Input is"

("WBUS 25 H" & "BUS CYC DEC L") +
("CUR MODE 1 H" & "BUS CYC DEC H")

"SAVED MODE 0 H" LATCH Input is"

("WBUS 24 H" & "BUS CYC DEC L") +
("CUR MODE 0 H" & "BUS CYC DEC H")

The latches are DC cleared when "PROC INIT H" is TRUE.

*"ENA WBUS H" is TRUE if:

("PHASE 1 L" & "SC ADD 3 L" & "SC ADD 2 L") & [WBUS <27:24> <- STATUS/CONTROL REGISTER(WCTRL Decode)]

*"PA WBUS H" is TRUE if:

("INVAL CHECK L" & "STATUS VAL L" & "CYC IN PROG L" & "PREFETCH L") & ("WRITE TB H" + "CA INVAL H" + "PTE CHECK H")

*"PA BUS DATA H" is TRUE if:

("STATUS VAL L" & "CYC IN PROG L" & "PREFETCH L") &
("WRITE TB H" + "PTE CHECK H")

*"CLK WDR H" IS TRUE IF:

("RTUT DINH H" & "ADD REG ENA H" &
"PREFETCH L" & "LATCHED BUS 3 L") +
"M CLK ENABLE H" & [2A + 2E(WCTRL Decodes)]

*"CLK MDR H" is TRUE if:

"D CLK ENABLE H" & [23 + 27 + 2B + 2F(WCTRL Decodes)]
+ [("CYC IN PROG H" & "STATUS VAL L" &
"PREFETCH DEL L" & "SCND REF L) &
("BUS GRANT DEC H" + "ADD REG ENA L" + "D CLK ENABLE
H") & ("PHASE 1 L" + "ADD REG ENA L") &
("READ H" + "BUS GRANT DEC H")]

*"SCND REF H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:(Bus Function Decodes)

READ LOCK TIMEOUT TEST +
READ, SECOND REFERENCE

*"FORCE DBUS CMI H" is TRUE if:

("ADD REG ENA L" & "READ H" & "CYC IN PROG H") +
("BUS GRANT DEC H" & "PREFETCH DEL L" & "PHASE 1 L")
+ ("STATUS VAL H" & "ADD ENA DEL L" & "READ H")

*"DBUS SEL S1 H" is TRUE if:

"PA BUS DATA H" +
("CYC IN PROG H" & "ADD REG ENA L" & "READ L") +

"FORCE DBUS CMI L" & ("ADD REG ENA L" + "READ L") &
("ADD REG ENA H" + "BUS CYC DEC L") &
("CACHE CYC L" + "PREFETCH DEL L") &
[23 + 26 + 27 + 28 + 29 + 2A + 2B + 2E + 2F(WCTRL Decodes)]
*"DBUS SEL S0 H" is TRUE if:

"FORCE DBUS CMI H" +

("ADD REG ENA L" + "READ L") &
("ADD REG ENA H" + "BUS CYC DEC L") &
("CACHE CYC L" + "PREFETCH DEL L") &
[26 + 27 + 2B + 2F(WCTRL Decodes)]

FLIP-FLOP DEFINITONS: All flip-flops are clocked on the RISING edge of "B CLK L".

#"LATCHED BUS 4 H - JK FLIP-FLOP
J INPUT:

"M CLK ENABLE H" & "BUS 4 H"

K INPUT:

"M CLK ENABLE H" & "BUS 4 L"

#"PREFETCH DEL H - D FLIP-FLOP
D INPUT:

"PREFETCH H"

The flop is DC cleared if:

"PREFETCH L" + "PROC INIT H"

#"PREFETCH CYC H - D FLIP-FLOP
D INPUT:

"PREFETCH DEL H"

The flop is DC PRESET when "PREFETCH DEL H" is TRUE.

#"UTRAP DEL H - D FLIP-FLOP
D INPUT:

"UTRAP H"

The flop is DC cleared when "UTRAP H" is FALSE.

#"ADD REG ENA H - JK FLIP-FLOP
J INPUT:

"MEM REQ H" & "INVAL CHECK L"

K INPUT:

"PREFETCH CYC H" + "M CLK ENABLE H"

#"ADD ENA DEL H - D FLIP-FLOP
D INPUT:

"ADD REG ENA H"

The flop is DC PRESET when "ADD REG ENA H" is TRUE.

#"INVAL CHECK H - JK FLIP-FLOP
J INPUT:

("MMUX SEL S1 L" & "SNAPSHOT CMI H") &
[("ADD REG ENA H" & "RESET ADD ENA H") + "MEM REQ L"]

K INPUT:

    "SNAPSHOT CMI L"

0"REPLACE 0 H - D FLIP-FLOP
D INPUT:

    ("TB CTL 3 H" & "TB CTL 2 L") +
    ("TB CTL 3 L" & "REPLACE 0 L")

The flop is DC PRESET wher. "PROC INIT H" is TRUE.

CHIP OUTPUTS

>>> "TB HIT 1 H" (OPEN COLLECTOR, BI-DIRECTIONAL) is driven LOW if:

"TB CTL 1 H"

>>> "TB HIT 0 H" (OPEN COLLECTOR, BI-DIRECTIONAL) is driven LOW if:

"TB CTL 0 H"

>>> "WBUS 27 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

"ENA WBUS H"

and is TRUE (HIGH) if:

("SC ADD 1 H" & "SC ADD 0 H" & "TB CTL 3 H") +
("SC ADD 1 L" & "SC ADD 0 H" & "SAVED MODE 3 H")

>>> "WBUS 26 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

"ENA WBUS H"

and is TRUE (HIGH) if:

("SC ADD 1 H" & "SC ADD 0 H" & "TB CTL 2 H") +
("SC ADD 1 L" & "SC ADD 0 H" & "SAVED MODE 2 H")

>>> "WBUS 25 H"(TRI-STATE, BI-DIRECTIONAL) is enabled if:

"ENA WBUS H"

and is TRUE (HIGH) if:

("SC ADD 1 H" & "SC ADD 0 H" & "TB CTL 1 H") +
("SC ADD 1 L" & "SC ADD 0 H" & "SAVED MODE 1 H")

```
>>>   "WBUS 24 H"(TRI-STATE, BI-DIRECTIONAL) is enabled if:

      "ENA WBUS H"

and is TRUE (HIGH) if:

      ("SC ADD 1 H" & "SC ADD 3 H" & "TB CTL 0 H") +
      ("SC ADD 1 H" & "SC ADD 0 L" & "WR VECT H") +
      ("SC ADD 1 L" & "SC ADD 0 H" & "SAVED MODE 0 H") +
      ("SC ADD 1 L" & "SC ADD 0 L" & "MME H")

>>>   "AMUX SEL S1 H" is TRUE (HIGH) if:

      "PA WBUS H" + "PHYS ADD H"

>>>   "AMUX SEL S0 H" is TRUE (HIGH) if:

      "INVAL CHECK H" + "PA BUS DATA H"

>>>   "BSRC SEL S1 H" is TRUE (HIGH) if:

      ("LATCHED WCTRL 3 L" & "LATCHED WCTRL 2 L" & "PHASE 1 L")

>>>   "BSRC SEL S0 H" is TRUE (HIGH) if:

      "PHASE 1 H" +
      ("LATCHED WCTRL 3 H" & "LATCHED WCTRL 0 L") +
      ("LATCHED WCTRL 1 H" & "LATCHED WCTRL 0 L")
>>>   "CLK SEL S1 H" is TRUE (HIGH) if:

      "CLK WDR H" +
      ("PREFETCH DEL H" & "STATUS VAL L")

>>>   "CLK SEL S0 H" is TRUE (HIGH) if:

      "CLK WDR H" + "CLK MDR H"

>>>   "COMP MODE H" is TRUE (HIGH) if:

      "PSL CM H" & ("PREFETCH H" + "FULL ADD L")

>>>   "DBUS SEL S1 H" is TRUE (HIGH) if:

      "DBUS SEL S1 H"

>>>   "DBUS SEL S0 H" is TRUE (HIGH) if:

      "DBUS SEL S0 H" + "FORCE DBUS CMI H"

>>>   "ENA VA L" is TRUE (LOW) if:

      "DEST VA H" & "D CLK ENABLE H"
```

```
>>>   "PTE CHECK L" is TRUE (LOW) if:

      ("PTE CHECK H" + "WRITE TB H") & "PREFETCH L"
>>>   "TB GRP 1 WR H" is TRUE (HIGH) if:

      ["INVAL CHECK L" & 29(WCTRL Decode)] +
      "D CLK ENABLE H" & 28(WCTRL Decode) &
      ["LATCHED HIT 1 H" + ("LATCHED HIT Ø L" & "REPLACE Ø L")]

>>>   "TB GRP Ø WR H" is TRUE (HIGH) if:

      ["INVAL CHECK L" & 29(WCTRL Decode)] +
      "D CLK ENABLE H" & 28(WCTRL Decode) &
      ["LATCHED HIT Ø H" + ("LATCHED HIT 1 L" & "REPLACE Ø H")]

>>>   "TB OUTPUT ENA L" is TRUE (LOW) if:

      "INVAL CHECK L" & "PA WBUS L" & "PHYS ADD L" & "PA BUS
      DATA L"
      ["ADD REG ENA H" +
      ("DBUS SEL S1 L" & "DBUS SEL SØ L" & "FORCE DBUS CMI L")]
>>>   "TB PARITY ENA H" is TRUE (HIGH) if:

      "MME H" &
      ["PREFETCH DEL H" + ("BUS GRANT DEC L" & "PHYS DEC L")]
```

```
                      ---------------
    AMUX SEL S1 H <---!01         48!<--- B CLK L
    DBUS SEL S1 H <---!02         47!---> AMUX SEL S0 H
        WBUS 26 H <-->!03         46!<--- DST RMODE H
        WBUS 25 H <--->!04        45!<--- MMUX SEL S1 H
        WBUS 24 H <-->!05         44!<--- WRITE VECT OCC L
        WBUS 27 H <-->!06         43!<--- SNAPSHOT CMI L
    TB GRP 0 WR H <---!07         42!<--- PSL CM H
       TB HIT 0 H <--->!08        41!<--- RTUT DINH L
    TB GRP 1 WR H <---!09         40!<--- STATUS VALID L
       TB HIT 1 H <-->!10 ....... 39!<--- M CLK ENABLE H
         ENA VA L <---!11 .     . 38!---- GROUND
              VGA ----!12 . LID . 37!<--- D CLK ENABLE H
              VCC ----!13 . DOWN. 36!<--- BUS 4 H
LATCHED WCTRL 2 H --->!14 .     . 35!---- GROUND
    BSRC SEL S0 H <---!15 ....... 34!---> TB PARITY ENA H
LATCHED WCTRL 5 H --->!16         33!<--- LATCHED BUS 3 H
LATCHED WCTRL 0 H --->!17         32!<--- LATCHED BUS 0 H
LATCHED WCTRL 3 H --->!18         31!---> PTE CHECK L
    BSRC SEL S1 H <---!19         30!<--- LATCHED BUS 1 H
LATCHED WCTRL 1 H --->!20         29!<--- PREFETCH L
LATCHED WCTRL 4 H --->!21         28!---> COMP MODE H
     CLK SEL S0 H <---!22         27!<--- LATCHED BUS 2 H
     CLK SEL S1 H <---!23         26!---> TB OUTPUT ENA L
        PHASE 1 H --->!24         25!---> DBUS SEL S0 H
                      ---------------
```

ADK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.4 ARITHMETIC AND LOGIC CONTROL CHIP, (ALK-DC615)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements of an Arithmetic and Logic Control Chip (ALK). The ALK chip accomplishes the following:

A) Reencodes the ALP control field (ALPCTL) for some special functions.

B) Controls the carry input and shift inputs for the ALP chip.

C) Decodes the scratch pad write enable signals.

D) Decodes some miscellaneous signals.

ALK TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	QSIO 07 L	I_{OL} = 12mA	GA1TCG
2	ASIO 00 L	I_{OL} = 12mA	GA1TCG
3	ASIO 31 L	I_{OL} = 12mA	GA1TCF
4	ALPCTL 0 H		GA1TNF
5	ALPCTL 1 H		GA1TNF
6	ALPCTL 6 H		GA1TNF
7	ROT 3 H		GA1TNF
8	ROT 4 H		GA1TNF
9	ROT 2 H		GA1TNF
10	DOUBLE ENABLE H		GA1TTN
11	PSL C H		GA1TNF
14	OPCODE 6 H		GA1TTN
15	ROT 0 H		GA1TNF
16	OPCODE 5 H		GA1TTN
17	ROT 1 H		GA1TNF
18	QD CLK L		GA1TNF
19	ROT 5 H		GA1TNF
20	ALPCTL 7 H		GA1TNF
21	CARRY OUT L		GA1TTN
22	ALPCTL 9 H		GA1TNG
23	OPCODE 4 H		GA1TTN
24	OPCODE 1 H		GA1TTN
25	BYTE L		GA1TTN
26	D SIZE 1 H		GA1TNF
27	SPWW EN H		GA1TTN
28	SPWL EN H		GA1TTN
29	SPWB EN H		GA1TTN
30	SPW 0 H		GA1TNF
31	D SIZE 0 H		GA1TNF
32	SPW 1 H		GA1TNF
33	LLIT L		GA1TNF
34	ALPCTL 3 H		GA1TNF
36	ALPCTL 2 H		GA1TNF
37	ALPCTL 5 H		GA1TNF
39	BCD L		GA1TTN
40	ALPCTL 4 H		GA1TNF
41	ALUC 31 L		GA1TNF
42	OPCODE 0 H		GA1TTN
43	ALPCTL 8 H		GA1TNF
44	QSIO 31 L	I_{OL} = 12mA	GA1TCF
45	QSIO 15 L	I_{OL} = 12mA	GA1TCG
46	WBUS 30 H		GA1TZG
47	WBUS 31 H		GA1TZG
48	QSIO 00 L	I_{OL} = 12mA	GA1TCG

2. Performance:

2.1 Special Signals and Mnemonics:

The ALK chip decodes two fields from the microword: the ALPCTL field and the ROT field. In the context of the ALP logic (the 8 ALP chips and the ALK chip), the ALPCTL field controls the main operations such as data input selection and ALU operation, whereas the ROT field controls the auxillary operation such as selecting carry inputs and shift-in inputs.

An exception occurs when both fields are used to represent a micro-code literal. This occurs when the "LLIT L" input is at a LOW level. Under this condition, the ALPCTL field and the ROT field lost their original meaning and are defaulted to a certain value as specified in section 2.4.

Throughout this document, mnemonics are used to represent some combinations of the ALPCTL field, and the mnemonics are meaningful only when the LLIT L input is at a HIGH level. Note that the mnemonics are not used to represent mutually exclusive conditions, for example, "Multiply" is a super-set of "Mul-", "Mul+", and "Mulfast".

Mnemonic	ALPCTL<9:6,5:2,1:0>	Short description
Div-	1001 1111 X0	Divide with negative dividend;
Div+	1001 1011 X0	Divide with positive dividend;
Mul-	1001 1010 X1	Multiply with negative multiplicand;
Mul+	1001 1110 X1	Multiply with positive multiplicand;
Mulfast	1001 1X10 01	Multiply at two iterations/cycle;
Divfast	1001 1X11 00	Divide at two iterations/cycle;
Multiply	1001 1X10 X1	Multiply functions;
Divide	1001 1X11 X0	Divide functions;
DIVD-	1001 1111 11	A function for divide double;
DIVD+	1001 1011 11	A function for divide double;
DIVD	1001 1X11 11	DIVD- or DIVD+;
REM	1001 1010 10	A function for remainder correction;
PASS S	1101 1X00 XX	Functions to pass Rotator through ALU;
ADD	XXXX 01XX XX	ALU operation is addition;
SUB	XXXX 00XX XX,or	
	XXXX 1100 XX	ALU operation is subtraction;
ALU SL	XXXX 0X11 XX,or	
	XXXX 1011 XX	ALU shift left;

ALU SR	XXXX 0X10 XX,or XXXX 1010 XX	ALU shift right;
ALU NS	.not.(ALU SL .or. ALU SR)	
Q SL	X0X1 XXXX X0	Q register shift left;
Q SR	X0X1 XXXX X1	Q register shift right;
Q NS	.not.(Q SL .or. Q SR)	
A Def	11X1 XXXX XX X11. XXXX XX,or X100 XXXX XX,or	A condition which indicates that the ALU is selecting the Rotator output. Under this condition, defaults are taken for the Shift out and the CARRY OUT L signals.

2.2 **Memory Elements:** The ALK chip has four D-type flip flops. When enabled, loading of the flip flops occurs at the rising edge of the QD CLK L clock signal.

ALK<C>: The ALK<C> is used to save the carry/borrow from ALU<31> every time an ADD/SUB operation is performed by the ALU:

LLIT L	ALPCTL	ALUC 31 L	ALK<C> Loaded with
H	ADD	L	1
H	ADD	H	0
H	SUB	L	0
H	SUB	H	1

In addition, on a Multiply operation, the ALK<C> is used to save the shift out from ALU<00>. Under all other conditions, ALK<C> is not loaded.

The ALK<C> can be sourced onto WBUS 30 H, see 3.3.4.4.

ALUSO: The ALUSO is used to save the shift out bit from the ALU on every ALU shift operation.

LLIT L	ALPCTL	ASIO 31 L	ASIO 00 L	ALUSO loaded
H	ALU SL	L	X	1
H	ALU SL	H	X	0
H	ALU SR	X	L	1
H	ALU SR	X	H	0

In addition, on a Divide operation, the shift out from ALU<31> is saved in ALUSO. Under all other conditions, ALUSO is not loaded.

The ALUSO can be sourced onto WBUS 31 H, see.

LOOP FLAG: This flop is used to enter a Multiply or Divide loop. It is loaded on every rising edge of the QD CLK L.

LLIT L	ALPCTL	LOOP FLAG loaded with
L	X	0
H	Multiply	1
H	Divide	1
H	'1011110XX	LOOP FLAG
H	all others	0

TOG FLAG: This flip flop is ɹsed by the ALK in the Multiply or Divide operations.

During a Divide operation, value related to the ALUC 31 L input is loaded into the TOG FLAG flop. This is used to control the OPCODE 1 H output.

During a Multiply operation, value related to the QSIO 00 L input is loaded into the TOG FLAG flop. This is used to control the OPCODE 4 H output.

ALPCTL	ALUC 31 L	QSIO 00 L	TOG FLAG loaded with
Multiply	X	H	0
Multiply	X	L	1
Div-	H	X	0
Div-	L	X	1
Div+	H	X	1
Div+	L	X	0

For all other conditions, the state of this flip flop is un-predictable.

2.3 Pin Summary: See Paragraph 2.4 for Pin Functions.

	Pin			
Name	I/O	Count	Electrical	Logical
ALPCTL <9:0> H	IN	10		Active Hi
LLIT L	IN	1		Active Lo
ROT <5:0> H	IN	6		Active Hi
WBUS <31:30> H	IN/OUT	2	Tri state	Active Hi
ASIO 31 L	IN/OUT	1	Open C	Active Lo
ASIO 00 L	IN/OUT	1	Open C	Active Lo
QSIO 31 L	IN/OUT	1	Open C	Active Lo
QSIO 15 L	IN/OUT	1	Open C	Active Lo
QSIO 07 L	IN/OUT	1	Open C	Active Lo
QSIO 00 L	IN/OUT	1	Open C	Active Lo
PSL C H	IN	1		Active Hi
ALUC 31 L	IN	1		Active Lo
SPW <1:0> H	IN	2		Active Hi
D SIZE <1:0> H	IN	2		Active Hi
SPWB EN L	OUT	1	TTL	Active Lo
SPWW EN L	OUT	1	TTL	Active Lo
SPWL EN L	OUT	1	TTL	Active Lo
QD CLK L	IN	1		Active Lo
OPCODE <6:4,1:0> H	OUT	5	TTL	Active Hi
DOUBLE ENABLE H	OUT	1	TTL	Active Hi
BCD L	OUT	1	TTL	Active Hi
CARRY OUT L	OUT	1	TTL	Active Lo
BYTE L	OUT	1	TTL	Active Lo

2.4 Pin Functions:

ALPCTL <9:0> H: A ten bit field from the microword to control the ALP and the ALK in general.

With the exception of the scratch pad write enable signals (SPWx EN L) and the BYTE L signal, all outputs from the ALK can be affected by this field.

ROT <5:0> H: A six bit field used by the ALK chip to control CARRY OUT L, ASIO <31,00> L, and QSIO <31,15,07,00> L pins.

LLIT L: This is a decoded signal from the LIT field in the microword. The signal is asserted when the LIT field specifies a long literal operation. Assertion of this signal would have special effects on the outputs and the flip flops, regardless of the other input conditions. These are summarized below:

OPCODE <6:4,1:0> H will be forced to LHHLL, this is used to effect a no shift on the ALU and a no-op on the Q and the D registers in the ALP chips.

ASIO <31,00> L, and QSIO <31,15,07,00> L will be forced to H;

DOUBLE ENAPLE H will be forced L;

BCD L will be forced H;

CARRY OUT L will be forced to L;

Output to WBUS <31:30> H will be forced to hi Z;

The two flops ALKC and ALUSO will remain intact;

The LOOP FLAG flop will be cleared.

The TOG FLAG flop will be unpredictable;

The BYTE L and the SPWx EN L signals are not affected.

WBUS <31:30> H: These are two bi-directional pins to WBUS <31:?0>.

As input, signals on the two pins are used as sources for ASIO <31,00> L and QSIO <31,00> L.

As output, the two pins are used to source the internal flip flops onto the WBUS:

LLIT L	ALPCTL<9:0>	WBUS<31>	WBUS<30>
L	X	Hi Z	Hi Z
H	1101111 1XX	ALUSO	ALKC
H	1131111 0XX	0	LOOP FLAG
H	All others	Hi Z	Hi Z

ASIO <31,00> L, QSIO <31,15,07,00> L: These are bi-directional shift input/output for the ALU and the Q register in the ALP chips. The exact output signal from these pins are defined in.

ASIO 31 L: During an ALU SL operation, this pin receives the signal shifted out from ALU<31> which is saved in the ALUSO flop.

During an ALU SR operation, this pin supplies the signal to be shifted into ALU<31>.

ASIO 00 L: During an ALU SR operation, this pin receives the signal shifted out from ALU<00> which is saved in the ALUSO flop.

During an ALU SL operation, this pin supplies the signal to be shifted into ALU<00>.

QSIO 31 L: During a Q SL operation, this pin receives the signal shifted out from Q register<31>.

During a Q SR operation, this pin supplies the signal to be shifted into Q register<31>.

QSIO 00 L: During a Q SR operation, this pin receives the signal shifted out from Q register<00>.

During A Q SL Operation, this pin supplies the signal to be shifted into Q register<00>.

QSIO 15 L and QSIO 07 L: These are used only for the Multiply and Divide operations:

LLIT L	ALPCTL	DSIZE	QSIO<15>	QSIO <07>
L	X	X	pull up	pull up
H	Multiply	LL(byte)	pull up	input
H	Multiply	LH(word)	input	pull up
H	Divide	LL(byte)	pull up	output
H	Divide	LH(word)	output	pull up

Under all other conditions, the two pins will be pulled high. Also see Table I for output definitions.

CARRY OUT L: An output to the carry input of the ALU. The signal output is defined in Tables VI and VII.

PSL C H: An input signal used as a source for the carry input and the shift input to the ALU.

ALUC 31 L: The carry output from ALU<31>. Is is saved by the ALK chip every time the ALU performs an ADD or SUB operation.

In addition, the signal is used during a Multiply or a Divide operation.

SPW <1:0> H: This is a two bit field to enable writing to the scratch pads.

See 3.3.4.11

D SIZE <1:0> H: These are two input signals from the micro-sequencer to indicate the data size in general:

D SIZE <1:0> H	Data size
LL	byte
LH	word
HL	long
HH	quad

SPWB EN L, SPWW EN L, SPWL EN L: These are three output signals to enable the scratch pad write operations.

When asserted, SPWB EN L allow a scratch pad location in the bit position <07:00> to be written.
When asserted, SPWW EN L allow a scratch pad location in the bit position <15:07> to be written.
When asserted, SPWL EN L allow a scratch pad location in the bit position <31:16> to be written.

All three signals are defined by the SPW and the DSIZE inputs:

SPW	DSIZE	SPWL	SPWW	SPWB
LL	XX	H	H	H
LH	LL	H	H	L
LH	LH	H	L	L
LH	HX	L	L	L
HL	XX	L	L	L
HH	XX	L	L	L

OPCODE <6:4, 1:0> H: Functions performed by the ALP chips are controlled by a 10 bit signal, the OPCODE <9:0> H. Of these ten control signals, OPCODE <6:4,1:0> H are output from the ALK chips.
Normally, the ALK chip passes ALPCTL <6:4,1:0> H directly to the ALP chips as OPCODE <6:4,1:0> H; under certain conditions, the ALK chip would re-encode these signals so that the ALK and the ALPs together would achieve some special functions.

BYTE L: A signal decoded from the D SIZE <1:0> inputs:

D SIZE <1:0>	BYTE L
LL	L
LH	H
HL	H
HH	H

This signal is used to enable sign/zero extend for the ALP chips in the <15:07> bit position.

BCD L: An output signal from the ALK indicating that the is performing a BCD add or subtract.

LLIT L	ALPCTL	BCD L
L	XXXXXXXXXX	H
H	XXXX0X01XX	L
H	Others	H

DOUBLE ENABLE H: The signal is asserted when the LOOP FLAG is set and that ALPCTL <9:0> H specifies a multiply or a divide operation.

LLIT L	ALPCTL	LOOP FLAG	DOUBLE ENABLE H
L	X	X	L
H	Mulfast	0	L
H	Mulfast	1	H
H	Divfast	0	L
H	Divfast	1	H
H	all else	1	L

QD CLK L: This is a clock input. When enabled, all flip flops are loaded on the rising edge of the clock input.

LLIT|L	REM	MUL-	MUL+	DIVI	LOOP FLAG	DIVD	DSI E1H	DSI E0H	ROT S:0H	ASIO 31 L	ASIO 00 L	QSIO 31 L	QSIO 15 L	QSIO 07 L	QSIO 00 L	D	Z	Z
1	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0			
0	1	X	X	X	X	X	X	X	0	ALUSO	0	0	0	0	0			
0	X	1	X	X	0	X	X	X	0	0	0	0	0	0	0			
0	X	X	1	X	0	X	X	X	0	0	0	0	0	0	0			
0	X	1	X	X	1	X	0	0	0	0	0	0	0	ALU<00>		0		
0	X	1	X	X	1	X	C	1	0	0	0	0	ALU<00>		0	0		
0	X	1	X	X	1	X	1	X	0	0	0	ALU<00>	0	0	0			
0	X	X	1	X	1	X	0	0	0	ALUC 31 L		0	0	0	ALU<00>			0(0
	X	X	1	X	1	X	0	1	0	ALUC 31L		0	0	ALU<00>		0		0
0	X	X	1	X	1	X	1	X	0	ALUC 31 L		0	ALU<00>		0	0		0
0	X	X	X	1	X	X	0	0	0	0	Q<07>	0	0	0	ALUC 31 L			
0	X	X	X	1	X	X	0	1	0	0	Q<15>	0	0	0	ALUC 31 L			
0	X	X	X	1	X	X	1	X	0	0	Q<31>	0	0	0	ALUC 31 L			
0	X	X	X	X	X	1	X	X	0	0	ALU SO		0	0	0	ALUC 31		
L																		
0	0	0	0	0	X	0	X	X	X	(*)	(*)	(**)	0	0	(**)			

CONDITION

OUTPUT

ALK CHART 1

NOTATIONS

ALUSO = ALUSO flip flop
for the
ALU<00> = Input from ASIO 00 L
values
Q<n> = Input from QSIO n L
values.
X = Don't care or mutually
exluded condtions.
(*) = See ALK Chart 2
(**) = See ALK Chart 3

Signal defintions

Shift I/O pins. All

listed are logical

Signals are expressed in
Logical values.

A DEF	R DEF	ALU NS	ALU SR	ALU SL	ROT 4 H	ROT 3 H	ROT 2 H	ASIO 31 L	ASIO 00 L
1	X	X	X	X	X	X	X	0	0
X	1	X	X	X	X	X	X	0	0
X	X	1	X	X	X	X	X	0	0
0	0	X	1	X	0	0	0	0	0
0	0	X	1	X	0	0	1	1	0
0	0	X	1	X	0	1	0	SEE ALK CHART 4	0
0	0	X	1	X	0	1	1	SEE ALK CHART 4	0
0	0	X	1	X	1	0	0	0	0
0	0	X	1	X	1	0	1	1	0
0	0	X	1	X	1	1	0	WBUS 30 H	0
0	0	X	1	X	1	1	1	PSL C H	0
0	0	X	X	1	0	0	0	0	0
0	0	X	X	1	0	0	1	0	1
0	0	X	X	1	0	1	0	0	SEE ALK CHART 4
0	0	X	X	1	0	1	1	0	SEE ALK CHART 4
0	0	X	X	1	1	0	0	0	0
0	0	X	X	1	1	0	1	0	1
0	0	X	X	1	1	1	0	0	WBUS 30 H
0	0	X	X	1	1	1	1	0	PSL C H
CONDITIONS								OUTPUTS	

NOTATIONS:

X = DON'T CARE OR MUTUALLY
EXCLUEDED CONDITIONS

R DEF = ROT <5:0> H = $(39,45,47,59,61,63)_{10}$

ALK CHART 2
SIGNAL DEFINTIONS FOR THE
ALU SHIFT I/O PINS
THE CHART IS A CONTINUATION
FROM ALK CHART 1

A DEF	R DEF	Q NS	Q SR	Q SL	ROT 4 H	ROT 3 H	ROT 2 H	QSIO 31 L	QSIO 00 L
1	X	X	X	X	X	X	X	0	0
X	1	X	X	X	X	X	X	0	0
X	X	1	X	X	X	X	X	0	0
0	0	X	1	X	0	0	0	0	0
0	0	X	1	X	0	0	1	1	0
0	0	X	1	X	0	1	0	SEE CHART 4	0
0	0	X	1	X	0	1	1	SEE CHART 4	0
0	0	X	1	X	1	0	0	1	0
0	0	X	1	X	1	0	1	0	0
0	0	X	1	X	1	1	0	WBUS 30 H	0
0	0	X	1	X	1	1	1	PSL C H	0
0	0	X	X	1	0	0	0	0	0
0	0	X	X	1	0	0	1	0	1
0	0	X	X	1	0	1	0	0	SEE ALK CHART 4
0	0	X	X	1	0	1	1	0	SEE ALK CHART 4
0	0	X	X	1	1	0	0	0	1
0	0	X	X	1	1	0	1	0	0
0	0	X	X	1	1	1	0	0	WBUS 30 H
0	0	X	X	1	1	1	1	0	PSL C H

CONDITIONS OUTPUTS

NOTATIONS:

X = DON'T CARE OR MUTUALLY
EXCLUDED CONDITIONS

R DEF = ROT <5:0> H = $(39,45,47,59,61,63)_{10}$

ALK CHART 3
SIGNAL DEFINITIONS FOR THE
Q REGISTER SHIFT I/O PINS
THE CHART IS A
CONTINUATION FROM ALK CHART 1

ROT 2 H	ALU SL	ALU SR	ALU NS	Q SL	Q SR	Q NS	ASIO 31 L	ASIO 00 L
0	1	X	X	1	X	X	0	Q<31>
0	1	X	X	X	1	X	0	Q<00>
0	1	X	X	X	X	1	0	Q<31>
0	X	1	X	1	X	X	Q<31>	0
0	X	1	X	X	1	X	0	0
0	X	1	X	X	X	1	Q<31>	0
0	X	X	1	1	X	X	0	0
0	X	X	1	X	1	X	0	0
0	X	X	1	X	X	1	0	0
1	1	X	X	1	X	X	0	Q<31>
1	1	X	X	X	1	X	0	Q<00>
1	1	X	X	X	X	1	0	ALU<31>
1	X	1	X	1	X	X	Q<31>	0
1	X	1	X	X	1	X	Q<00>	0
1	X	1	X	X	X	1	ALU<00>	0
1	X	X	1	1	X	X	0	0
1	X	X	1	X	1	X	0	0
1	X	X	1	X	X	1	0	0

CONDTIONS OUTPUTS

NOTATIONS:

X = DON'T CARE OR MUTUALLY
EXCLUDED CONDITIONS.
Q<n> = INPUT FROM QSIO <n> L
ALU<n> = INPUT FROM ASIO <n> L

ALK CHART 4
SIGNAL DEFINITIONS FOR THE
ALU SHIFT I/O PINS
THE TABLE IS A CONTINUATION
FROM ALK CHART 2

ROT 2 H	ALU SL	ALU SR	ALU NS	Q SL	Q SR	Q NS	QSIO 31 L	QSIO ØØ L
Ø	1	X	X	1	X	X	Ø	Ø
Ø	1	X	X	X	1	X	Ø	Ø
Ø	1	X	X	X	X	1	Ø	Ø
Ø	X	1	X	1	X	X	Ø	Ø
Ø	X	1	X	X	1	X	ALU<ØØ>	Ø
Ø	X	1	X	X	X	1	Ø	Ø
Ø	X	X	1	1	X	X	Ø	WBUS 31 H
Ø	X	X	1	X	1	X	WBUS 31 H	Ø
Ø	X	X	1	X	X	1	Ø	Ø
1	1	X	X	1	X	X	Ø	ALU<31>
1	1	X	X	X	1	X	ALU<31>	Ø
1	1	X	X	X	X	1	Ø	Ø
1	X	1	X	1	X	X	Ø	ALU<ØØ>
1	X	1	X	X	1	X	ALU<ØØ>	Ø
1	X	1	X	X	X	1	Ø	Ø
1	X	X	1	1	X	X	Ø	Q<31>
1	X	X	1	X	1	X	Q<31>	Ø
1	X	X	1	X	X	1	Ø	Ø

CONDITIONS OUTPUTS

NOTATIONS:

X = DON'T CARE OR MUTUALLY
EXCLUDED CONDTIONS.
Q<n> = INPUT FROM QSIO <n> L
ALU <n> = INPUT FROM ASIO <n> L

ALK CHART 4
SIGNAL DEFINITIONS FOR THE
Q REGISTER SHIFT I/O PINS
THE CHART IS A CONTINUATION
FROM ALK CHART 3

LLIT L	REM	MUL-	MUL+	DIV-	DIV+	LOOP FLAG	DIV-	DIV+	PASS S	ROT 5:Ø H	CARRY OUT L
1	X	X	X	X	X	X	X	X	X	X	1
Ø	1	X	X	X	X	X	X	X	X	Ø	1
Ø	X	1	X	X	X	X	X	X	X	Ø	1
Ø	X	X	1	X	X	X	X	X	X	Ø	Ø
Ø	X	X	X	1	X	Ø	X	X	X	Ø	Ø
Ø	X	X	X	1	X	1	X	X	X	Ø	$\overline{TOG}$
Ø	X	X	X	X	1	Ø	X	X	X	Ø	1
Ø	X	X	X	X	1	1	X	X	X	Ø	$\overline{TOG}$
Ø	X	X	X	X	X	X	1	X	X	Ø	ALK<C>
Ø	x	x	x	x	x	x	x	1	x	Ø	$\overline{ALK<C>}$
Ø	X	X	X	X	X	X	X	X	1	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	X	Ø	Ø	Ø	X	SEE ALK CHART 6

CONDITION OUTPUT

NOTATIONS:

X = DON'T CARE OR MUTUALLY
EXCLUDED CONDITIONS.
TOG = TOG FLAG FLIP FLOP.
ALK<C> = ALK<C> FLIP FLOP.

ALK CHART 5
SIGNAL DEFINTIONS FOR THF
CARRY OUT L PIN
ALL VALUES LISTED ARE LOGICAL VALUES

LLIT L	ADEF	RDEF	ADD	ROT 1 H	ROT 0 H	CARRY OUT L
1	X	X	X	0	0	1
1	X	X	X	0	1	1
1	X	X	X	1	0	1
1	X	X	X	1	1	1
0	1	X	0	X	X	1
0	1	X	1	X	X	0
0	X	1	0	X	X	1
0	X	1	1	X	X	0
0	0	0	0	0	0	1
0	0	0	0	0	1	$\overline{\text{ALK<C>}}$
0	0	0	0	1	0	0
0	0	0	0	1	1	$\overline{\text{PSL<C>}}$
0	0	0	1	0	0	0
0	0	0	1	0	1	ALK<C>
0	0	0	1	1	0	1
0	0	0	1	1	1	PSL<C>

CONDITIONS OUTPUT

NOTATIONS:
X = DON'T CARE OR MUTUALLY
EXCLUDED CONDITIONS.
ALK<C> = THE ALK<C> FLIP FLOP.
PSL<C> = INPUT FROM PSL C H
R DEF ROT <5:0> H = $(39,45,47,59,61,63)_{10}$

ALK CHART 6
SIGNAL DEFINITIONS FOR THE
CARRY OUT L PIN
THE CHART IS A CONTINUATION
FROM ALK CHART 5
ALL VALUES ARE LOGICAL VALUES

LLIT L	REM	MUL-	MUL+	DIV-	DIV+	LOOP FLAG	DIVD-	DIVD+	PASS S	OPCODE <n> H				
										<6>	<5>	<4>	<1>	<0>
1	X	X	X	X	X	X	X	X	X	0	1	1	0	0
0	1	X	X	X	X	X	X	X	X	0	0	0	1	0
0	X	1	X	X	X	0	X	X	X	1	1	0	1	1
0	X	1	X	X	X	1	X	X	X	1	0	0	$\overline{TOG}$	1
0	X	X	1	X	X	0	X	X	X	1	1	0	1	1
0	X	X	1	X	X	1	X	X	X	1	0	1	$\overline{TOG}$	1
0	X	X	X	1	X	0	X	X	X	1	0	1	0	0
0	X	X	X	1	X	1	X	X	X	1	0	TOG	0	0
0	X	X	X	X	1	0	X	X	X	1	0	0	0	0
0	X	X	X	X	1	1	X	X	X	1	0	TOG	0	0
0	X	X	X	X	X	X	1	X	X	1	0	1	0	0
0	X	X	X	X	X	X	X	1	X	1	0	0	0	0
0	X	X	X	X	X	X	X	X	1	1	0	-	-	-
0	0	0	0	0	0	X	0	0	0	-	-	-	-	-

CONDITIONS OUTPUTS

NOTATIONS:
X = DON'T CARE OR MUTUALLY EXCLUDED CONDITIONS

- = OUTPUT SAME AS THE CORRESPONDING ALPCTL <n> H

TOG = TOG FLAG FLOP

ALK CHART 7
SIGNAL DEFINTIONS
FOR THE OPCODE <n> H PINS

```
                     ---------------
      QSIO 00 L <-->!01              48!<--> QSIO 00 L
```

```
                     ---------------
      QSIO 07 L <-->!01              48!<--> QSIO 00 L
      ASIO 00 L <-->!02              47!<--> WBUS 31 H
      ASIO 31 L <-->!03    ALK       46!<--> WBUS 30 H
     ALPCTL 0 H --->!04              45!<--> QSIO 15 L
     ALPCTL 1 H --->!05              44!<--> QSIO 31 L
     ALPCTL 6 H --->!06              43!<--- ALPCTL 8 H
        ROT 3 H --->!07              42!---> OPCODE 0 H
        ROT 4 H --->!08              41!<--- ALUC 31 L
        ROT 2 H --->!09              40!<--- ALPCTL 4 H
DOUBLE ENABLE H <---!10 .......      39!---> BCD L
        PSL C H --->!11 .     .      38!---- GROUND
            VGA ----!12 . LID .      37!<--- ALPCTL 5 H
            VCC ----!13 . DOWN.      36!<--- ALPCTL 2 H
     OPCODE 6 H <---!14 .     .      35!---- GROUND
        ROT 0 H --->!15 .......      34!<--- ALPCTL 3 H
     OPCODE 5 H <---!16              33!<--- LLIT L
        ROT 1 H --->!17              32!<--- SPW 1 H
       QD CLK L --->!18              31!<--- D SIZE 0 H
        ROT 5 H --->!19              30!<--- SPW 0 H
     ALPCTL 7 H --->!20              29!---> SPWB EN H
    CARRY OUT L <---!21              28!---> SPWL EN H
     ALPCTL 9 H --->!22              27!---> SPWW EN H
     OPCODE 4 H <---!23              26!<--- D SIZE 1 H
     OPCODE 1 H <---!24              25!---> BYTE L
                       ---------------
```

ALK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.5 ARITHMETIC LOGICAL PROCESSOR (ALP-DC608)

1. GENERAL DESCRIPTION:

The ALP chip performs the majority of the data manipulations in executing the macro instructions. Each chip is four bits wide and contains a conventional ALU, two D-type registers, shift networks, and BCD-adjust circuit.

ALP Table 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	SB 2 H		GA1TNG
2	QR SHF DATA0 L	I_{OL} = 12mA	GA1TCH
3	SB 1 H		GA1TNG
4	QD CLK L		GA1TNG
5	GROUP GENERATE L		GA1TTG
6	CARRY IN L		GA1TNF
7	ALU OVERFLOW H		GA1TTN
8	ALU SHF DATA3 L	I_{OL} = 12mA	GA1TCG
9	GROUP PROPAGATE L		GA1TTG
10	WB 3 H		GA1TZN
11	WB 1 H		GA1TZN
14	WB 2 H		GA1TZN
15	WB 0 H		GA1TZN
16	ALU SHF DATA0 L	I_{OL} = 12mA	GA1TCF
17	OPCODE 2 H		GA1TNG
18	OPCODE 4 H		GA1TNG
19	OPCODE 5 H		GA1TNG
20	OPCODE 3 H		GA1TNG
21	WMUX ZERO H	I_{OL} = 12mA	GA1TCN
22	OPCODE 6 H		GA1TNG
23	OPCODE 0 H		GA1TNG
24	OPCODE 1 H		GA1TNG
25	MB EXTEND EN L		GA1TNG
26	OPCODE 8 H		GA1TNG
27	OPCODE 7 H		GA1TNG
28	MB 1 L		GA1TNF
29	OPCODE 9 H		GA1TNG
30	MB 0 L		GA1TNF
31	MB 3 L		GA1TNF
32	MB 2 L		GA1TNF
33	RB 0 L		GA1TNF
34	LATCH EN L		GA1TNG
36	RB 1 L		GA1TNF
37	RB 2 L		GA1TNF
39	RB 3 L		GA1TNF
40	QR SHF DATA3 L	I_{OL} = 12mA	GA1TCI
41	SB 0 H		GA1TNG
42	SB 4 H		GA1TNG
43	SB 5 H		GA1TNG
44	SHF POS 00 L		GA1TNG
45	SB 3 H		GA1TNG
46	SB 6 H		GA1TNG
47	SHF POS 01 L		GA1TNG
48	EXT DATA L		GA1TNH

2 Performance Requirements:

2.1 Data Ports: Three input ports and one output port are provided for interfacing with other components in the CPU.

The M-port is used to accept data from the MBUS. The R-port is used to accept data from the RBUS. Data appearing on both buses are active low. Internal latches are provided to latch the data.
The S-port is used to accept data form the SBUS. Data appearing on this bus are active high. Unlike the MBUS and the RBUS, no internal latches are provided to latch the data.

The W-port is used to send data to the WBUS. Data appearing on this bus are active Hi.

2.2 CLOCKS: Two clock signals are used by the ALP chip:

a. LATCH EN L
b. QD CLK L.

The LATCH EN L is used to latch data from the M and R BUSes. When LATCH EN L is at a low level, input data is fed through the internal latch. At the rising edge of LATCH EN L, input data is retained until LATCH EN L goes low again. Inputs that are latched include MB <3:0> L, RB <3:0> L, and EXT DATA L. The QD CLK L is used to load the is internal registers (the Q and D registers.). When enabled, data is loaded at the rising edge of the clock.

2.3 Major Components: There are ten major sections to the ALP logic:

1. Input latches
2. S-Shifter
3. ALU input mux
4. ALU
5. BCD adjust
6. Output mux
7. Q-Register
8. D-Register
9. Function control
10. Status logic

With the exception of the S-Shifter, the control of all sections are decoded out of a 10 bit signal, "OPCODE <9:0> H". 1024 possible functions are shown in the ALP CTL FUNCTION CHART (This chart is provided in the DPM schematic package.).

Input Latches: Data coming from the RBUS and the MBUS are complemented and latched in the ALP chip using the "LATCH EN L" signal. Feed through latches are used so that data can be operated on at the same time they are being latched.

S-Shifter: The S-shifter is a combinatorial shifter capable of right shifting the S-port by 0, 1, 2, 3 bits. The number of bits to be shifted is specified as a two bit number, SHF POS <01:00> L:

SHF POS <1:0> L	SHIFT THE S-PORT BY
H H	0 bit
H L	1 bit right
L H	2 bits right
L L	3 bits right

ALU INPUT MUX: The A and B inputs to the ALU are controlled by the A-mux and B-mux respectively. The control for both mux's are decoded out of the OPCODE <9:0> H and the MB extend EN L inputs.

ALU: The ALU performs three binary arithmetic operations, two quasi-BCD arithmetic operations, and five logical operations.
The three binary arithmetic operations are:

```
A plus B plus Cin   (A + B + Cin)
A plus .not.B plus Cin (A - B - .NOT.Cin)
B plus .not.A plus Cin (B - A - .NOT.Cin)
```

In this mode, two carry look ahead signals (P and G) are calculated based on 16.

The two quasi-BCD arithmetic operations are:

```
A plus B plus Cin (A + B + Cin, BCD)
A plus .not.B plus Cin (A - B - .NOT.Cin, BCD)
```

In this mode, the output of the ALU is the same as doing binary arithmetic, but the P and G signals are calculated based on 10. Extra logic is used to adjust the 4 bit ALU output to a true BCD result.

The five logical operations are:

```
A.AND.B
A.OR.B
A.ANDNOT.B
B.ANDNOT.A
A.XOR.B
```

BCD Adjust: When the ALP opcode specifies a BCD operation, the output of the ALU may be adjusted to a legal BCD digit according to the following rules:

```
For A plus B plus Cin (Add)-
if the true sum is ten(10) or greater
    then BCD Adjust <- six(6) plus ALU output
    else BCD Adjust <- ALU output;

For A plus .not.B plus Cin (Subtract)-
if the true difference is less than zero (0)
    then BCD Adjust <- ten(10) plus ALU output
    else BCD Adjust <- ALU output
```

OUTPUT MUX (W-MUX): Data coming out of the ALP chip is controlled by W-MUX. There are five sources to this mux:

a. BCD adjust
b. ALU output
c. ALU shifted left
d. ALU shifted right
e. B-MUX output.

The BCD adjust is selected for a BCD arithmetic operation. Each 4 bit ALU output is adjusted to fall within the range of 0 through 9. (See description on BCD adjust logic).
When ALU shift left is selected, data to be shifted into the least significant bit appears at the ALU SHF DATA0 L pin, and ALU<3> will be shifted out to the ALU SHF DATA3 L pin.
When ALU shift right is selected, shift in and shift out data will appear at the ALU SHF DATA3 L and the ALU SHF DATA0 L pins, respectively.

When the B-MUX is selected, the ALU function is still defined by OPCODE <5:2> H, even though its output cannot be read by any logic. (But all status signals can be.)

Q-REGISTER: The Q-Register is a 4 bit D-type register.

Sources to the Q-register are either from the A-MUX or the W-MUX outputs. In addition, the Q-register can be shifted left or right by 1 bit.
On a shift left, the shift-in to the least significant bit is supplied by the QR SHF DATA0 L pin, and the most significant bit shifted out to the QR SHF DATA3 L pin.
On a shift right, the shift-in and the shift-out appears at the QR SHF DATA3 L and the QR SHF DATA0 L pins respectively.

D-Register: The D-Register is a 4 bit D-type register used to hold the intermediate results of the ALU. Source to the L-register is from the W-MUX output only.

FUNCTION CONTROL: All ALP slices receive a 10 bit OPCODE <9:0> H inputs. A total of 1024 functions are possible, these are tabulated in the ALP function (see DPM schematic package). In the ALP function Chart, there are 16 major columns. Each column is identified by OPCODE <9:6> H, which specifies the A and B inputs to the ALU. There are also 16 major rows. Each row is identified by OPCODE <5:2> H, which specifies the ALU and W-MUX operation. At the intersection of a major column and a major row, there are four blocks which are identified by OPCODE <1:0> H. Each block specifies an operation on the Q and D registers with the given ALU inputs and the ALU operation.

The following notations are used in the chart:

M : Data accepted from MBUS;
R : Data accepted from RBUS;
XM : Extended MBUS data, which is generated by replicating data accepted from the EXT DATA L pin.
XM is selected by the A-MUX if (OPCODE <9:6> H = 5, 6, or 7) and (MB EXTEND EN L = L). If MB EXTEND EN L is at a high level, A-MUX would select M instead;
S : S-shifter output;
A : A-MUX;
B : B-MUX;
WX : W-MUX;
CI : Carry input, (data on the CARRY IN L pin);
D : D-Register;
Q : Q-Register;
DSL : D-Register shifted left through the ALU;
DSR : D-Register shifted right through the ALU;
SQL : Q-Register shifted left;
SQR : Q-Register shifted right;
0SR : Zero shifted right through the ALU;

Except when OPCODE <9, 7:5> H = HLHH, data selected by the W-MUX are also output to the WB <3:0> H pins. When OPCODE <9, 7:5> H = HLHH, the WB <3:0> H pins are turned off.

STATUS LOGIC: Two status signals are available for condition code primitive or micro branches.

The "WMUX ZERO H" signal indicates whether the W-MUX output is all zero. This signal is valid even though output to the WBUS is disabled.

The "ALU OVERFLOW H" signal indicates an overflow condition detected by the ALP slices:

For binary arithmetic operations-
overflow <-- the exclusive OR of the carry-in and the carry-out of the most significant bit.

for logical operations-
overflow <-- 0

For BCD arithmetic operations-
overflow <-- 0

2.4 Pin Identifications, Pin Count and Descriptions: Per ALP Table 2.

2.5 Truth Table for "Group Propagate L" (A + B + C_i, BCD): Per ALP Table 3.

2.6 Truth Table for "Group Generate L" (A + B + C_i, BCD): Per ALP Table 4.

2.7 ALP Functions: See DPM Schematic Package.

ALP TABLE 2

PIN DEFINITIONS

Name	Pin Count	Description
MB <3:0> L	4	Accepts input data from the M-BUS.
RB <3:0> L	4	Accepts input data from the R-BUS.
SB <6:0> H	7	Accepts input data from the Rotator output. Four pins are for data and the remaining three for shift-in data.
SHF POS <01:00> L	2	Two input pins to control the number of bits the SBUS input is to be shifted. SHF POS<1:0> — Number of Bits to Be Shifted H H — 0 bits right H L — 1 bits right L H — 2 bits right L L — 3 bits right
EXT DATA L	1	The data input for sign/zero extension of the MBUS.
MB EXTEND EN L	1	Used to enable sign/zero extend of the MBUS input. When this signal is asserted, and that the ALP opcode pins specifies a zero/sign extend operation, the EXT DATA latch would be selected to the A leg of the ALU.
LATCH EN L	1	A signal used to latch data on the M and R BUSes.
QD CLK L	1	A clock signal used to edge trigger the Q and D registers.
OPCODE <9:0> H	10	Ten input pins to control all ALP components except the S-Shifter. See ALP CTL function chart (DPM schematic package).

ALP TABLE 2 (CONT.)

PIN DEFINITIONS

Name	Pin Count	Description
WB <3:0> H	4	Used to send data to the W-BUS. These four pins can be switched to a hi impedance state by a specific set of opcode combinations indicated in the ALP function table.
GROUP PROPAGATE L	1	The group carry propagate output used for carry look ahead.

ALU operation	Signal is asserted if
Binary Subtract	A = B
BCD Subtract	A = B
Binary Add	A + B = F (hex)
BCD Add	See ALP Table 3
Logical functions	No useful definition.

GROUP GENERATE L	1	The group carry generate output used for carry look ahead.

ALU operation	Signal is asserted if
Binary Subtract (A-B)	A > B
BCD Subtract	A > B
Binary Subtract (B-A)	B > A
Binary Add	A + B > F (hex)
BCD Add	See ALP Table 4
Logical functions	No useful definition.

WMUX ZERO H	1	This pin indicates whether the output mux is all zero. This signal is valid even though the ALU output is turned off.
ALU OVERFLOW H	1	An output pin indicating an overflow condition.
CARRY IN L	1	Carry input for binary and BCD arithmetic.
ALU SHF DATA0 L	1	On a shift left, this pin accepts data to be shifted into ALU<0>. On a shift right, this pin transmits the data shifted out of ALU<0>.

ALP TABLE 2 (Cont.)

PIN DEFINITIONS

Name	Pin Count	Description
ALU SHF DATA3 L	1	On a shift right, this pin accepts data to be shifted into ALU<3>. On a shift left, this pin carries the data shifted out of the ALU<3>.
QR SHF DATA0 L	1	On a shift left, this pin accepts data to be shifted into Q<0>. On a shift right, this pin carries data shifted out of the Q<0>.
QR SHF DATA3 L	1	On a shift right, this pin accepts data to be shifted into Q<3>. On a shift left, this pin carries data shifted out of Q<3>.

ALP TABLE 3

Truth Table for "GROUP PROPAGATE L" (A + B + Ci, BCD)

A= \ B=	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	H	H	H	H	H	H	H	H	H	L	H	L	H	L	H	L
1	H	H	H	H	H	H	H	H	L	H	L	H	L	H	L	H
2	H	H	H	H	H	H	H	L	H	L	H	L	H	L	H	L
3	H	H	H	H	H	H	L	H	L	H	L	H	L	H	L	H
4	H	H	H	H	H	L	H	L	H	L	H	L	H	L	H	L
5	H	H	H	H	L	H	L	H	L	H	L	H	L	H	L	H
6	H	H	H	L	H	L	H	L	H	L	H	L	H	L	H	L
7	H	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
8	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
9	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
10	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
11	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
12	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
13	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H
14	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L
15	L	H	L	H	L	H	L	H	L	H	L	H	L	H	L	H

ALP TABLE 4

Truth Table for "GROUP GENERATE L" (A + B + Ci, BCD)

\B= A=\	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	H	H	H	H	H	H	H	H	H	H	L	L	L	L	L	L
1	H	H	H	H	H	H	H	H	H	L	L	L	L	L	L	L
2	H	H	H	H	H	H	H	H	L	L	L	L	L	L	L	L
3	H	H	H	H	H	H	H	L	L	L	L	L	L	L	L	L
4	H	H	H	H	H	H	L	L	L	L	L	L	H	H	L	L
5	H	H	H	H	H	L	L	L	L	L	L	L	H	L	L	L
6	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L
7	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L
8	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L
9	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
10	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
11	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
12	L	L	L	L	H	H	L	L	L	L	L	L	L	L	L	L
13	L	L	L	L	H	L	L	L	L	L	L	L	L	L	L	L
14	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
15	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

```
                                  --------------
               SB 2 H --->|01                48|<--- EXT DATA L
     QR SHF DATA0 L <-->|02                47|<--- SHF POS 01 L
               SB 1 H --->|03     ALP       46|<--- SB 6 H
            QD CLK L --->|04                45|<--- SB 3 H
  GROUP GENERATE L <---|05                44|<--- SHF POS 00 L
         CARRY IN L --->|06                43|<--- SB 5 H
    ALU OVERFLOW H <---|07                42|<--- SB 4 H
    ALU SHF DATA3 L <-->|08                41|<--- SB 0 H
 GROUP PROPAGATE L <---|09                40|<--> QR SHF DATA3 L
               WB 3 H <---|10 .......   39|<--- RB 3 L
               WB 1 H <---|11 .           . 38|---- GROUND
                    VGA ----|12 . LID .   37|<--- RB 2 L
                    VCC ----|13 . DOWN. 36|<--- RB 1 L
               WB 2 H <---|14 .           . 35|---- GROUND
               WB 0 H <---|15 .......   34|<--- LATCH EN L
    ALU SHF DATA0 L <-->|16                33|<--- RB 0 L
         OPCODE 2 H --->|17                32|<--- MB 2 L
         OPCODE 4 H --->|18                31|<--- MB 3 L
         OPCODE 5 H --->|19                30|<--- MB 0 L
         OPCODE 3 H --->|20                29|<--- OPCODE 9 H
        WMUX ZERO H <---|21                28|<--- MB 1 L
         OPCODE 6 H --->|22                27|<--- OPCODE 7 H
         OPCODE 0 H --->|23                26|<--- OPCODE 8 H
         OPCODE 1 H --->|24                25|<--- MB EXTEND EN L
                                  --------------
```

ALP FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION
(SEE PARA. 2.4 (ALP TABLE 2) FOR PIN DEFINITIONS AND DESCRIPTIONS)

3.6 CACHE ADDRESS CONTROL CHIP (CAK-DC627)

1. **GENERAL DESCRIPTION:**
The CAK chip generates most of the signals necessary to control the cache and contains the status/control registers associated with the cache.

CAK TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	MA ØØ H		GA1TNF
2	WBUS 27 H		GA1TZF
3	B CLK L		GA1TNF
4	WBUS 26 H		GA1TZF
5	HIT 1 H		GA1TCF
6	WBUS 25 H		GA1TZF
7	CACHE INT L		GA1TTN
8	WBUS 24 H		GA1TZF
9	HIT Ø H	I_{oL} = 2ØmA	GA1TCF
1Ø	CACHE VALID 1 H		GA1TTN
11	SNAPSHOT CMI L		GA1TNF
14	DATA PAR ERR L		GA1TNF
15	TAG Ø PAR ERR H		GA1TNF
16	TAG 1 PAR ERR H		GA1TNF
17	CACHE GRP 1 WR H		GA1TTN
18	CACHE GRP Ø WR H		GA1TTN
19	ENA BYTE Ø L		GA1TTN
2Ø	CACHE VALID Ø H		GA1TTN
21	ENA BYTE 3 L		GA1TTN
22	ENA BYTE 1 L		GA1TTN
23	ENA BYTE 2 L		GA1TTN
24	MMUX SEL S1 H		GA1TNF

CAK TABLE 1 (Cont)

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
25	MA Ø1 H		GA1TNF
26	DBUS ROT S1 H		GA1TTN
27	IO ADDRESS L		GA1TNF
28	M CLK ENABLE H		GA1TNF
29	PREFETCH L		GA1TNF
3Ø	STATUS VALID L		GA1TNF
31	D SIZE 1 H		GA1TNF
32	D SIZE Ø H		GA1TNF
33	DST RMODE H		GA1TNF
34	D CLK ENABLE H		GA1TNF
36	LATCHED WCTRL 5 H		GA1TNF
37	LATCHED WCTRL 1 H		GA1TNF
39	LATCHED WCTRL 3 H		GA1TNF
4Ø	LATCHED BUS 3 H		GA1TNF
41	LATCHED BUS 4 H		GA1TNF
42	LATCHED BUS 2 H		GA1TNF
43	LATCHED BUS Ø H		GA1TNF
44	LATCHED BUS 1 H		GA1TNF
45	LATCHED WCTRL 2 H		GA1TNF
46	LATCHED WCTRL 4 H		GA1TNF
47	LATCHED WCTRL Ø H		GA1TNF
48	DBUS ROT SØ H		GA1TTN

2. **Performance:**

2.1 CAK I/O PINS shall be as specified herein.

INPUTS	#PINS
B CLK L	1
D CLK ENABLE H	1
D SIZE <1:0> H	2
DATA PAR ERR L	1
DST RMODE H	1
IO ADDRESS L	1
LATCHED BUS <4:0> H	5
LATCHED WCTRL <5:0> H	6
M CLK ENABLE H	1
MAD <01:00> H	2
MMUX SEL S1 H	1
PREFETCH L	1
SNAPSHOT CMI L	1
STATUS VALID L	1
TAG <1:0> PAR ERR H	2

BI-DIRECTIONAL LINES (TRI-STATE UNLESS OTHERWISE NOTED)

HIT <1:0> H	2(OPEN-COLL.)
WBUS <27:24> H	4

OUTPUTS (TOTEM POLE UNLESS OTHERWISE NOTED)

CACHE INT L	1
CACHE GRP <1:0> WR H	2
CACHE VALID <1:0> H	2
DBUS ROT S<1:0> H	2
ENA BYTE <3:0> L	4

2.2 The following is a list of the code assignments for the BUS FUNCTION Micro field:

0. READ PHYSICAL ADDRESS
1. PROCESSOR INITIALIZE
2. READ, NO MICRO-TRAP
3. I/O INITIALIZE
4. READ LOCK TIMEOUT TEST
5. NOP
6. READ, SECOND REFERENCE
7. NOP
8. WRITE PHYSICAL ADDRESS
9. REI CHECK
A. WRITE, SECOND REFERENCE
B. WRITE UNLOCK, SECOND REFERENCE
C. WRITE, NO MICRO-TRAP
D. NOP
E. WRITE LONGWORD, NO MICRO-TRAP
F. BUS GRANT
10. READ
11. READ LONGWORD
12. PTE ACCESS CHECK, WRITE
13. READ LOCK
14. READ WITH MODIFY INTENT
15. READ LONGWORD WITH MODIFY INTENT
16. PTE ACCESS CHECK, READ
17. PTE ACCESS CHECK, READ, KERNEL MODE
18. WRITE
19. WRITE LONGWORD
1A. WRITE IF NOT RMODE
1B. WRITE UNLOCK
1C. PROBE ACCESS, WRITE, MODE SPECIFIED
1D. PROBE ACCESS, WRITE
1E. PROBE ACCESS, READ, MODE SPECIFIED
1F. PROBE ACCESS, READ

Bus Functions are decoded from "LATCHED BUS <4:0> H.

2.3 The following is a list of the code assignments for the upper half of the WCTRL Micro field:

20.	VA <- PC + ISIZE + (WBUS)
	PC <- PC + ISIZE

21.	VA <- VA SAVE + (WBUS)
22.	VA <- VA + 4
23.	MDR <- (WBUS)
24.	PC <- (WBUS)
25.	VA <- (WBUS)
26.	MBUS <- WDR
27.	MDR <- 0
28.	TB DATA <- (WBUS)
29.	TB VALID BIT <- 0 VA <- (WBUS)
2A.	WDR <- (WBUS) UNROTATED
2B.	MDR <- IR, ZERO EXTENDED
2C.	PC <- PC + (WBUS)
2D.	CACHE VALID BIT <- 0 VA <- (WBUS)
2E.	WDR <- (WBUS)
2F.	MDR <- OSR, ZERO EXTENDED
30.	STATUS/CONTROL REGISTER <- WBUS<27:24>
31.	PREVIOUS MODE REGISTER <- WBUS<23:22>
32.	WBUS<27:24> <- STATUS/CONTROL REGISTER
33.	BUS GRANT
34.	STATUS/CONTROL ADDRESS REGISTER <- WBUS<27:24>
35.	IS/CURRENT MODE REGISTER <- WBUS<26:24>
37.	REI CHECK
38.	ASTLVL REGISTER <- WBUS<26:24>
39.	(RESERVED)
3A.	WBUS<26:24> <- ASTLVL REGISTER
3B.	(RESERVED)
3C.	HIGHEST SOFTWARE IPR REGISTER <- WBUS<20:16>
3D.	IPL REGISTER <- WBUS<20:16>
3E.	NOP
3F.	WBUS<20:16> <- IPL OF HIGHEST IPR

WCTRL Functions are decoded from "LATCHED WCTRL <5:0> H"

2.4 Internal Signal Definitions:

*"BUS CYC DEC H" is TRUE for any of the following Bus Function Decodes:

READ PHYSICAL ADDRESS
READ, NO MICRO-TRAP
READ LOCK TIMEOUT TEST
READ, SECOND REFERENCE
WRITE PHYSICAL ADDRESS
WRITE, SECOND REFERENCE
WRITE UNLOCK, SECOND REFERENCE
WRITE, NO MICRO-TRAP
WRITE LONGWORD, NO MICRO-TRAP
BUS GRANT
READ
READ LONGWORD
READ LOCK
READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
WRITE
WRITE LONGWORD
WRITE UNLOCK

"DST RMODE L" & WRITE IF NOT RMODE

*"BUS GRANT DEC H" is TRUE for: (Bus Function Decode)

BUS GRANT

*"PROC INIT H" is TRUE if:

"PHASE 1 L" & PROCESSOR INITIALIZE (Bus Function Decode)

*"PAR ERR 0 H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"TAG 0 PAR ERR H"

*"DATA PERR H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"DATA PAR ERR H"

*"LATCHED HIT 1 H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"HIT 1 H"

*"LATCHED HIT 0 H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"HIT 0 H"

*"LATCHED MA 01 H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"MAD 01 H"

*"LATCHED MA 00 H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is:

"MAD 00 H"

*"SIZE 1 H" is the HIGH TRUE output of a latch which is enabled during:

"SIZE LATCH ENA H"

LATCH Input is:

"D SIZE 1 H"

*"SIZE 0 H" is the HIGH TRUE output of a latch which is enabled during:

"SIZE LATCH ENA H"

LATCH Input is:

"D SIZE 0 H"

*"SIZE LATCH ENA H" is TRUE if:

"ADD REG ENA H" & "PREFETCH DEL L" &
[WRITE + WRITE LONGWORD + WRITE UNLOCK +
WRITE IF NOT RMODE (Bus Function Decodes)]

*"ENA CACHE ERR H" is TRUE if:

"ADD REG ENA H" & "RESET ADD ENA H" &
["PREFETCH DEL H" + ("D CLK ENABLE H" & "BUS GRANT DEC
L")]

*"RESET ADD ENA H" is TRUE if:

"PREFETCH CYC H" + "M CLK ENABLE H"

*"STEER TAG ERR H" is TRUE if:

"ENA CACHE ERR H" &
["TAG 1 PAR ERR H" +
("PAR ERR 0 H" & "HIT 0 H" & "READ H") +
("LATCHED HIT 1 H" & "LATCHED HIT 0 H")]

*"STEER DATA ERR H" is TRUE if:

"ENA CACHE ERR H" & "DATA PERR H" &
"HIT 0 H" & "READ H"

*"ENA LAST REF H" is TRUE if:

"ENA CACHE ERR H" & "PREFETCH DEL L"

*"RESET TAG ERR H" is TRUE if:

"TAG ERROR L" & "B CLK L"

>>NOTE: "RESET TAG ERR H" does NOT "Glitch" TRUE after the RISING edge of "B CLK L" which is causing the "TAG ERROR H" Register bit to become set.

*"RESET DATA ERR H" is TRUE if:

"DATA ERROR L" & "B CLK L"

>>NOTE: "RESET DATA ERR H" does NOT "Glitch" TRUE after the RISING edge of "B CLK L" which is causing the "DATA ERROR H" Register bit to become set.

*"ENA WBUS H" is TRUE if:

("PHASE 1 L" & "SC ADD 3 L" & "SC ADD 2 H") &
[WBUS <27:24> <- STATUS/CONTROL REGISTER (WCTRL Decode)]

*"CACHE INVAL H" is TRUE if:

2D (WCTRL Decode)

*"WRITE CACHE H" is TRUE if:

```
        "IO ADDRESS L" &
        [("WRITE ALLOCATE H" & "ENA ALLOCATE H" & "INVAL CHECK
        L") +
        ("STATUS VAL H" & "READ H" & "ADD ENA DEL L")]
```

*"READ H" is the HIGH TRUE output of a latch which is enabled during:

```
        "ADD REG ENA H"
```

LATCH Input is:

```
        "PREFETCH DEL H" + "LATCHED BUS 3 L"
```

*"ENA ALLOCATE H" is TRUE if:

```
        ("WR CACHE ONLY H" + "STATUS VAL L") &
        ["LATCHED HIT 1 H" + "LATCHED HIT 0 H" + "ALL BYTES H" +
        ("LATCHED MA 01 L" & "LATCHED MA 00 L" & "SIZE 1 H")]
```

*"ALL BYTES H" is the HIGH TRUE output of a latch which is enabled during:

```
        "ADD REG ENA H"
```

LATCH Input is TRUE if "PREFETCH H" is TRUE or for any of the following Bus Function Decodes:

READ PHYSICAL ADDRESS
WRITE PHYSICAL ADDRESS
BUS GRANT
READ, NO MICRO-TRAP
READ, SECOND REFERENCE
WRITE LONGWORD
WRITE LONGWORD, NO MICRO-TRAP READ
READ LONGWORD
READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
READ LOCK

*"READ ROT DEC H" is TRUE for any of the following Bus Function Decodes:

READ
READ WITH MODIFY INTENT
READ LOCK
READ, SECOND REFERENCE
READ, NO MICRO-TRAP

*"LATCHED READ ROT H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

"READ ROT DEC H" & "D CLK ENABLE H" & "B CLK H"

and reset by:

"STATUS VAL H"

*"LD WDR ROT H" is TRUE if:

2E (WCTRL Decode)

*"CYC IN PROG H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

"ADD REG ENA H"

and reset by:

"STATUS VAL H" + "PROC INIT H"

*"WRITE SCND H" is the HIGH TRUE output of a latch which is enabled during:

"ADD REG ENA H"

LATCH Input is TRUE for: (Bus Function Decodes)

WRITE, SECOND REFERENCE
WRITE UNLOCK, SECOND REFERENCE

*"MEM REQ H" is TRUE if:

["PREFETCH DEL H" +
("BUS CYC DEC H" & "PREFETCH L" & "REPLACEMENT L")] &
("ADD REG ENA H" + "CYC IN PROG L")

*"REPLACEMENT H" is TRUE if:

"STATUS VAL H" & "READ H" & "ADD ENA DEL L"

*"ENABLE INVAL H" is TRUE if:

"MMUX SEL S1 L" &
["MEM REQ L" + ("ADD REG ENA H" & "RESET ADD ENA H")]

*"INVAL HIT 0 H" is the HIGH TRUE output of a latch which is enabled during:

"INVAL WRITE L"

LATCH Input is:

"HIT 0 H"

S/C ADDRESS REGISTER

The S/C ADDRESS REGISTER is a 4 bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
STATUS/CONTROL ADDRESS REGISTER <- WBUS <27:24> (WCTRL Decode)

LATCH Inputs are: "WBUS <27:24> H"
LATCH Outputs are: "SC ADD <3:0> H"

STATUS/CONTROL REGISTERS

CACHE ERROR REGISTER

The CACHE ERROR REGISTER is a 4 bit edge triggered register which is clocked on the RISING edge of "B CLK L" if:

"D CLK ENABLE H" &
"SC ADD 3 L" & "SC ADD 2 H" & "SC ADD 1 L" & "SC ADD 0 L" &
STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)

REGISTER Inputs are: "WBUS <27:24> H" REGISTER Outputs <3:0> are:

Bit <3>: "TAG ERROR H"
Bit <2>: "DATA ERROR H"
Bit <1>: "LOST ERROR H"
Bit <0>: "LAST REF HIT H"

In addition, the individual bits may by SYNCHRONOUSLY (on the RISING edge of "B CLK L") PRESET and/or CLEARED as follows:

B t <3> PRESET if:

"STEER TAG ERR H"

Bit <2> PRESET if:

"STEER DATA ERR H"

Bit <1> PRESET if:

("TAG ERROR H" + "DATA ERROR H") &
("STEER TAG ERR H" + "STEER DATA ERR H")

Bit <0> PRESET if:

"ENA LAST REF H" & "HIT 0 H"

Bit <0> CLEARED if:

"ENA LAST REF H" & "HIT 0 L"

All 4 bits are ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

CACHE GROUP DISABLE REGISTER

The CACHE GROUP DISABLE REGISTER is a 4 bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 L" & "SC ADD 2 H" & "SC ADD 1 H" & "SC ADD 0 L" &
STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)

LATCH Inputs are: "WBUS <27:24> H"
LATCH Outputs are: "CACHE CTL <3:0> H"

All 4 bits are ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

WRITE CACHE ONLY REGISTER

The WRITE CACHE ONLY REGISTER is a one bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 H" & "SC ADD 2 H" & "SC ADD 1 H" & "SC ADD 0 L" &
STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)

LATCH Input is: "WBUS 24 H"
LATCH Output is: "WR CACHE ONLY H"

The latch is ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

FLIP-FLOP DEFINITIONS
All flip-flops are clocked on the RISING edge of "B CLK L".

#"STATUS VAL H" - D FLIP-FLOP
D INPUT:

"STATUS VALID H"

The flop is DC CLEARED when "STATUS VALID H" is FALSE.

#"PREFETCH DEL H" - D FLIP-FLOP
D INPUT:

"PREFETCH H"

The flop is DC cleared if "PREFETCH H" is FALSE.

#"PREFETCH CYC H" - D FLIP-FLOP

D INPUT:

"PREFETCH DEL H"

The flop is DC PRESET when "PREFETCH DEL H" is TRUE.

#"ADD REG ENA H" - JK FLIP-FLOP

J INPUT:

"MEM REQ H" & "INVAL CHECK L"

K INPUT:

"PREFETCH CYC H" + "M CLK ENABLE H"

#"ADD ENA DEL H" - D FLIP-FLOP

D INPUT:

"ADD REG ENA H"

The flop is DC PRESET when "ADD REG ENA H" is TRUE.

#"INVAL CHECK H" - JK FLIP-FLOP

J INPUT:

"SNAPSHOT CMI H" & "ENABLE INVAL H"

K INPUT:

"SNAPSHOT CMI L"

#"INVAL WRITE H" - D FLIP-FLOP

D INPUT:

"INVAL CHECK H"

The flop is DC CLEARED when "INVAL CHECK H" is FALSE.

#"WRITE ALLOCATE H" - JK FLIP-FLOP

J INPUT:

"ENA CACHE ERR H" & "READ L"

K INPUT:

"INVAL CHECK L" + "ENA ALLOCATE L"

#"CACHE INT H" - JK FLIP-FLOP

J INPUT:

"ENA CACHE ERR H" & ("STEER TAG ERR H" + "STEER DATA ERR H")

K INPUT:

"M CLK ENABLE H"

The flop is CLEARED when "PROC INIT H" is TRUE.

#"REPLACE 0 H" - D/TOGGLE FLIP-FLOP

If "CACHE CTL 3 H" is TRUE, the flop is a D FLIP-FLOP.
D INPUT:

"CACHE CTL 2 L"

When "CACHE CTL 3 H" is FALSE, the flop toggles (on the RISING edge of "B CLK L") if "WRITE CACHE H" is TRUE.

The flop is PRESET when "PROC INIT H" is TRUE.

#"PHASE 1 H" - D FLIP-FLOP
D INPUT:

"M CLK ENABLE H"

CHIP OUTPUTS

>>> "HIT 1 H" (OPEN-COLLECTOR, BI-DIRECTIONAL) is driven LOW if:

"CACHE CTL 1 H"

>>> "HIT 0 H" (OPEN-COLLECTOR, BI-DIRECTIONAL) is driven LOW if:

"CACHE CTL 0 H"

>>> "WBUS 27 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

"ENA WBUS H"

and is TRUE (HIGH) if:

("SC ADD 1 H" & "SC ADD 0 L" & "CACHE CTL 3 H") +
("SC ADD 1 L" & "SC ADD 0 L" & "TAG ERROR H")

>>> "WBUS 26 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

"ENA WBUS H"

and is TRUE (HIGH) if:

("SC ADD 1 H" & "SC ADD 0 L" & "CACHE CTL 2 H") +
("SC ADD 1 L" & "SC ADD 0 L" & "DATA ERROR H")

>>> "WBUS 25 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

```
     "ENA WBUS H"

and is TRUE (HIGH) if:

     ("SC ADD 1 H" & "SC ADD 0 L" & "CACHE CTL 1 H") +
     ("SC ADD 1 L" & "SC ADD 0 L" & "LOST ERROR H")

>>> "WBUS 24 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

     "ENA WBUS H"

and is TRUE (HIGH) if:

     ("SC ADD 1 H" & "SC ADD 0 L" & "CACHE CTL 0 H") +
     ("SC ADD 1 L" & "SC ADD 0 L" & "LAST REF HIT H")

>>> "CACHE INT L" is TRUE (LOW) if:

     "CACHE INT H"

>>> "CACHE GRP 1 WR H" is TRUE (HIGH) if:

     "CACHE INVAL H" +

     ("INVAL WRITE H" & "HIT 1 H") +

     "WRITE CACHE H" &
     ["LATCHED HIT 1 H" + ("LATCHED HIT 0 L" & "REPLACE 0 L")]

>>> "CACHE GRP 0 WR H" is TRUE (HIGH) if:

     "WRITE CACHE H" +

     ("CACHE INVAL H" & "D CLK ENABLE H") +

     ("INVAL WRITE H" & "INVAL HIT 0 H")

>>> "CACHE VALID 1 H" is TRUE (HIGH) if:

     ("CACHE INVAL L" & "INVAL WRITE L") &
     [("LATCHED HIT 1 H" & "LATCHED HIT 0 L") +
     ("LATCHED HIT 1 H" & "REPLACE 0 L") +
     ("LATCHED HIT 0 L" & "REPLACE 0 L")]

>>> "CACHE VALID 0 H" is TRUE (HIGH) if:

     "CACHE INVAL L" & "INVAL WRITE L" &

     ["PAR ERR 0 L" + "READ H" + "ALL BYTES H" +
     ("SIZE 1 H" & "LATCHED MA 01 L" & "LATCHED MA 00 L")]

>>> "DBUS ROT S1 H" is TRUE (HIGH) if:

     ["LATCHED MA 01 H" &
     ("LATCHED READ ROT H" +
     "READ ROT DEC H" & "D CLK ENABLE H")] +

     [("LD WDR ROT H" & "M CLK ENABLE H") &
     ("MAD 01 H" .xor. "MAD 00 H")]

>>> "DBUS ROT S0 H" is TRUE (HIGH) if:

     ["LATCHED MA 00 H" &
     ("LATCHED READ ROT H" +
     "READ ROT DEC H" & "D CLK ENABLE H")] +

     ("LD WDR ROT H" & "M CLK ENABLE H" & "MAD 00 H") +

     [("READ L" + "CYC IN PROG L") &
     (26 + 27 (WCTRL Decodes))]

>>> "ENA BYTE 3 L" is TRUE (LOW) if:

     "ALL BYTES H" + "WRITE ALLOCATE L" +

     "WRITE SCND L" &
     ["SIZE 1 H" + ("LATCHED MA 01 H" & "LATCHED MA 00 H") +
     ("LATCHED MA 01 H" & "SIZE 0 H")]

>>> "ENA BYTE 2 L" is TRUE (LOW) if:

     "ALL BYTES H" + "WRITE ALLOCATE L" +

     "WRITE SCND L" &
     [("LATCHED MA 01 L" & "SIZE 1 H") +
     ("LATCHED MA 01 H" & "LATCHED MA 00 L") +
     ("LATCHED MA 01 L" & "LATCHED MA 00 H" & "SIZE 0 H")] +

     "WRITE SCND H" &
     ("LATCHED MA 01 H" & "LATCHED MA 00 H" & "SIZE 1 H")

>>> "ENA BYTE 1 L" is TRUE (LOW) if:

     "ALL BYTES H" + "WRITE ALLOCATE L" +

     [("WRITE SCND L" & "LATCHED MA 01 L") &
     ("LATCHED MA 00 H" + "SIZE 1 H" + "SIZE 0 H")] +

     "WRITE SCND H" &
     ("LATCHED MA 01 H" & "SIZE 1 H")

>>> "ENA BYTE 0 L" is TRUE (LOW) if:

     "ALL BYTES H" + "WRITE SCND H" + "WRITE ALLOCATE L" +
     ("LATCHED MA 01 L" & "LATCHED MA 00 L")
```

```
                        ---------------
          MA 00 H --->!01         48!---> DBUS ROT S0 H
        WBUS 27 H <-->!02         47!<--- LATCHED WCTRL 0 H
          B CLK L --->!03         46!<--- LATCHED WCTRL 4 H
        WBUS 26 H <-->!04         45!<--- LATCHED WCTRL 2 H
          HIT 1 H <-->!05         44!<--- LATCHED BUS 1 H
        WBUS 25 H <-->!06         43!<--- LATCHED BUS 0 H
      CACHE INT L <---!07         42!<--- LATCHED BUS 2 H
        WBUS 24 H <-->!08         41!<--- LATCHED BUS 4 H
          HIT 0 H <-->!09         40!<--- LATCHED BUS 3 H
  CACHE VALID 1 H <---!10 ....... 39!<--- LATCHED WCTRL 3 H
   SNAPSHOT CMI L --->!11 .     . 38!---- GROUND
              VGA ----!12 . LID . 37!<--- LATCHED WCTRL 1 H
              VCC ----!13 . DOWN. 36!<--- LATCHED WCTRL 5 H
   DATA PAR ERR L --->!14 .     . 35!---- GROUND
  TAG 0 PAR ERR H --->!15 ....... 34!<--- D CLK ENABLE H
  TAG 1 PAR ERR H --->!16         33!<--- DST RMODE H
 CACHE GRP 1 WR H <---!17         32!<--- D SIZE 0 H
 CACHE GRP 0 WR H <---!18         31!<--- D SIZE 1 H
     ENA BYTE 0 L <---!19         30!<--- STATUS VALID L
  CACHE VALID 0 H <---!20         29!<--- PREFETCH L
     ENA BYTE 3 L <---!21         28!<--- M CLK ENABLE H
     ENA BYTE 1 L <---!22         27!<--- IO ADDRESS L
     ENA BYTE 2 L <---!23         26!---> DBUS ROT S1,H
    MMUX SEL S1 H --->!24         25!<--- MA 01 H
                        ---------------
```

CAK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.7 CONDITION CODE CONTROL (CCC-610)

1. GENERAL DESCRIPTION:

The primary function of the Condition Code (CC) chip is to determine the condition codes for VAX and compatibility mode instructions, store the PSL bits <FU,IV,DV,N,Z,V,C> and read these bits to the data path at ucode request. The inputs to the N, Z, V, C are selected from data path primitives by internal decode of the opcode. Logic is included to signal the ucode in case of arithmetic traps. Two branch signals are provided for ucode branching. These are used to signal true branch conditions on branch instructions and to supply latched or unlatched data path condition flags.

CCC TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	D SIZE Ø H		GA1TNF
2	D SIZE 1 H		GA1TNH
3	ALUZ BØ H		GA1TNF
4	IR 6 H		GA1TNF
5	IR 4 H		GA1TNF
6	IR 7 H		GA1TNF
7	IR 5 H		GA1TNF
8	IR 3 H		GA1TNF
9	IR 2 H		GA1TNF
1Ø	IR 1 H		GA1TNF
11	IR Ø H		GA1TNF
14	FPA Z L		GA1TNF
15	ARITH TRAP L		GA1TTF
16	FPA V L		GA1TNF
17	PSL C H		GA1TTN
18	CCBR Ø H		GA1TTN
19	PROC INIT L		GA1TNF
2Ø	B CLK L		GA1TNF
21	WBUS ØØ H		GA1TZF
22	WBUS Ø3 H		GA1TZF
23	WBUS Ø1 H		GA1TZF
24	WBUS Ø2 H		GA1TZF
25	D CLK ENABLE H		GA1TNF
26	CC CTRL Ø H		GA1TNF
27	CC CTRL 2 H		GA1TNF
28	CC CTRL 1 H		GA1TNF
29	CC CTRL 3 H		GA1TNF
3Ø	WBUS Ø4 H		GA1TNF
31	ALUZ B2 H		GA1TNF
32	WBUS Ø5 H		GA1TZF
33	CCBR 1 H		GA1TTN
34	WBUS Ø6 H		GA1TZF
36	WBUS Ø7 H		GA1TZF
37	ALUZ B3 H		GA1TNF
39	WBUS 31 H		GA1TNF
4Ø	ALUC Ø7 H		GA1TNF
41	ALUC 15 H		GA1TNH
42	ALUC 31 H		GA1TNH
43	ALUZ B1 H		GA1TNF
44	ALUV 15 H		GA1TNF
45	ALUV Ø7 H		GA1TNF
46	ALUV 31 H		GA1TNF
47	WBUS 15 H		GA1TNF
48	FPA PRESENT L		GA1TPF

2.1 **Performance:**

CLOCKS - B CLK L, D CLK ENABLE H: B CLK L & D CLK ENABLE H are TTL inputs to CCC.

The internal signal, D CLK L is generated from the signals B CLK L and D CLK ENABLE H.

D CLK L = B CLK L + .NOT. D CLK ENABLE H.

2.2 CC CTRL <3:0> H: CC CTRL <3:0> H is the primary control field for the CCC chip. It controls the setting of all state elements in CCC, and the values sourced onto WBUS <7:5> H, WBUS <3:0> H, and CCBR <1:0> H.

D SIZE <1:0> H control multiplexers for the following sets of signals.

INPUTS	OUTPUT
WBUS <31,15,07> H	SIGN (SZ)
WMUXZ B <3:0> H	WMUX (SZ)
ALUV <31,15,07> H	OV (SZ)
ALUC <31,15,07> L	$\overline{\text{CRY}}$ (SZ)

These outputs are defined in Para. 2.7.

WBUS <31,15,7:0> H: This is a bidirectional tri-state port. (Note: WBUS <31,15,04> H are only inputs.)

The tri-state drivers are enabled as a function of CC CTRL <3:0> H.

If CC CTRL <3:0> H = 2
then WBUS <3:0> H <- ATCR <3:0>

If CC CTRL <3:0> H = 3
then WBUS <7:5> H <- PSL <FU,IV,DV>
WBUS <3:0> H <- PSL <N,Z,V,C>

COMP H: COMP H is a transparent latch loaded from WBUS <31> H. It is enabled when D CLK L is low and CC CTRL <3:0> H = 9.

CCC is in native (VAX) mode if COMP H = 0.

CCC is in compatability mode if COMP H = 1.

2.3 Condition Codes - PSL<N,Z,V,C>: The Condition Codes are modified in three ways, as selected by CC CTRL:

A. Directly loaded from WBUS nn H.
B. Modified by direct CC CTRL function
C. Modified as a function of Opcode.

Direct Load of Condition Codes: The condition codes are loaded from WBUS<3:0> for the following control states:

Function	CC CTRL (HEX)
Write PSL	9
Write CC	A
Write PSW	B

Direct Modification of Condition Codes: When CC CTRL = F, PSL<V> is set and PSL <N,Z,C> are unchanged.

Modification of Condition Codes by Opcode: When CC CTRL = C (CC OF OP 1) or E (CC OF OP 2), the Condition Codes are modified as a function of the following:

A. IR n H

B. COMP H

C. CC CTRL = C, CC of op 1
E, CC of op 2

D. D SIZE n H

E. FPA PRESENT L

F. Data Path primitives (see below)

G. FP Z L and FP V L

Paragraph 2.7 lists the Data Path primitives selected for native mode instructions (enabled when COMP H = 0) and compatibility mode instructions (enabled when COMP H = 1).

The lists in Paragraph 2.7 do not completely describe the operation of CCC for all values of IR n H and COMP H. Operation for any values not listed is undefined.

Condition Code Input Primitives: The following signals are input to CCC.

A. WMUXZ B0 H

B. WMUXZ B1 H

C. WMUXZ B2 H

D. WMUXZ B3 H

E. WBUS<07> H

F. WBUS<15> H

G. WBUS<31> H

H. ALUV 07 H

I. ALUV 15 H

J. ALUV 31 H

K. ALUC 07 L

L. ALUC 15 L

M. ALUC 31 L

FPA Present L - FP Z L, FP V L: Under certain conditions, FP Z L, and FP V L are used as the low true inputs to PSL <Z> and PSL <V> respectively.

FP Z L and FP V L are used to set PSL <Z, V> instead of the primitives specified in Paragraph 2.7 when the following conditions are true. PSL <N,C> are handled as in Para. 2.7.

A. COMP H = 0

B. FPA PRESENT L = 0

C. IR n H = FPA op

4. CC CTRL implies CC OF OP 1 or CC OF OP 2

Where FPA OP = 01x0xxxx
  01xx0x01
  01xx010x

PSL C H: PSL C H is a TTL output equal to PSL <C>.

2.4 FU, IV, DV: PSL<7:5> (DV, FU, IV) are transparent latches, open when D CLK L is low and CC CTRL <3:0> H = 9 or 8. They are read/write through WBUS<7:5> respectively. They are output to the WBUS when CC CTRL <3:0> H = 3.

ARITH TRAP L: ARITH TRAP L the inverted output of an edge triggered flip flop, set with the positive going edge of D CLK L. It is only cleared by the assertion of PROC INIT L or by D CLK L being low with CC CTRL = 2.

Arith Trap is set as a function of

A. CC CTRL

B. Input to PSL V

C. COMP H

D. IR n H

E. PSL <DV>

F. PSL (IV)

G. FPA PRESENT

H. FPA OP

SET ARITH TRAP = (TRAP CC CTRL) * (INPUT TO PSL V) *.NOT.[(FPA PRESENT) * (FPA OP)] * (COMP H =0) * [(DV * DV OP) +
(IV * IV OP)]

where

TRAP CC CTRL = (CC CTRL = A + C + E + F)

```
DV OP = (IR n H = 00xx10xx
                  + 111x10xx
                  + 00100xxx)
IV OP = (IR n H = 01x010xx
                  + 1xx0xxxx
                  + xxx10x00
                  + x10x1000
                  + 1x0xxxxx
                  + x0x10xxx
                  + 111x0xxx
                  + 0x1111xx
                  + 01111xxx)
```

Arichmetic Trap Code Register (ATCR): The ATCR is a four bit read only register which can only contain the value 0110 or 0001.

```
ATCR <-- 6 if (TRAP CC CTRL)*(DV OP) *(COMP H = 0)
ATCR <-- 1 if (TRAP CC CTRL)*(IV OP) *(COMP H = 0)
```

ATCR is changed with the negative going edge of D CLK L.

WBUS<3:0> <-- ATCR if CC CTRL = 2

The ATCR is not changed by PROC INIT or by reading.

2.5 CCBR <1:0> H: CCBR n H are two TTL outputs. There are three different ways in which these lines are driven.

A. For VAX or Compatability mode branch instructions test.

B. The outputs of the ALU STATE latches can be sourced onto these lines.

C. Unlatched status information can be sourced onto these lines.

CCBR - CC CTRL = 1: If CCC CTRL <3:0> H =1, CCBR 1 H = 0, and CCBR 0 H is defined herein:

IR <7:Ø> H	COMP H	CONDITIONS FOR CCBR Ø H = 1
12	*	Z=Ø
13	*	Z=1
14	Ø	(N+V)=Ø
14	1	(N●V)=Ø
15	Ø	(N+V)=1
15	1	(N●V)=1
16	1	(Z+(N●V))=Ø
17	1	(Z+(N●V))=1
18	*	N=Ø
19	*	N=1
1A	*	(C+Z)=Ø
1B	*	(C+Z)=1
1C	*	V=Ø
1D	*	V=1
1E	*	C=Ø
1F	*	C=1

CCBR Ø H is underfined for any other state of IR <7:Ø> H and COMP H.

ALU STATE<1:Ø>: The ALU STATE latches are edge triggered flip-flops that are clocked on the rising edge of the D CLK L if CC CTRL <3:Ø> H = 5 or 6 or 7.

If CC CTRL <3:Ø> H = 5,
ALU STATE 1 H <- if WBUS <3:Ø> H = 1 or 3 or 9 or B or D
else ALU STATE 1 H <- Ø

If CC CTRL <3:Ø> H = 7
ALU STATE 1 H <- CRY (SZ)
ALU STATE Ø H <- (WMUX (SZ) = Ø)

CCBR - CC CTRL = Ø, 2, 3, C, E, F: If CC CTRL <3:Ø> H = Ø or 2 or 3 or C or E or F,

CCBR 1 H = SIGN (SZ) ● OV (SZ)
CCBR Ø H = (WMUX (SZ) = Ø)

CCBR - CC CTRL = 4, 5, 6, 7, 9, A, B: If CC CTRL <3:Ø> H = 4 or 5 or 6 or 7 or 9 or A or B,

CCBR 1 H = ALU STATE 1 H
CCBR Ø H = ALU STATE Ø H
CCBR - CC CTRL = 8: If CC CTRL <3:Ø> H = 8,

CCBR 1 H = ALU STATE 1 H
CCBR Ø H = SIGN (SZ) ● OV (SZ)

CCBR - CC CTRL = D: If CC CTRL <3:Ø> H = 8,

CCBR <1:Ø> H are undefined.

2.6 PROC INIT L: When PROC INIT L is asserted low, the ARITH TRAP latch is asynchronously cleared.

2.7 Definition of Terms Used in CCC Table 2: SIGN(SZ) = WBUS<31> iff DSIZE = 11 or 1Ø

WBUS<15> iff DSIZE = Ø1
WBUS<Ø7> iff DSIZE = ØØ

WMUX(SZ) = Ø is determined by the WMUXZ B3 H, WMUXZ B2 H, WMUXZ B1 H, WMUXZ BØ H signals from the ALP. It equals:

WMUXZ B3 H * WMUXZ B2 H * WMUXZ B1 H * WMUXZ BØ H iff DSIZE = 11 or 1Ø

WMUXZ B1 H * WMUX BØ H iff DSIZE = Ø1
WMUXZ BØ H iff DSIZE = ØØ

OV(SZ) = ALUV<31> H iff DSIZE = 11 or 1Ø
ALUV<15> H iff DSIZE = Ø1
ALUV<Ø7> H iff DSIZE = ØØ

$\overline{\text{CRY}}$(SZ) = ALUC<31> L iff DSIZE = 11 or 1Ø
ALUC<15> L iff DSIZE = Ø1
ALUC<Ø7> L iff DSIZE = ØØ

$\overline{\text{CRY}}$(SZ) is the inverse of CRY(SZ)

+ is logical OR, * is logical AND, ⊕ is logical XOR

FP CMP N = WBUS<15> + [(WMUX[W]=Ø)*CRY(SIZE)]

CCC Table 2

Integer Arithmetic & Logical Instructions:

INSTR	N	Z	V	C	CC OF OP
MOVB MOVW MOVL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
MOVQ	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	C	1
	SIGN(SZ)	WMUX(SZ)=Ø *Z	Ø	C	2
PUSHL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
CLRB CLRW CLRL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
CLRQ	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	C	1
MNEGB MNEGW MNEGL	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	1
MCOMB MCOMW MCOML	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
MOVZBW MOVZBL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
MOVZWL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
CVTBW CVTBL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
CVTWL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
INSTR	N	Z	V	C	CC OF OP

CCC Table 2

Integer Arithmetic & Logical Instructions: (Cont)

CVTWB					
CVTLB	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
CVTLW	N	Z	WMUX[L] ≠Ø	C	2
CMPB	SIGN(SZ)				
CMPW	●	WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1
CMPL	OV(SZ)				
TSTB	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
TSTW					
TSTL					
ADDB2,3	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
ADDW2,3					
ADDL2,3					
INCB	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
INCW					
INCL					
ADWC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
SUBB2,3	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
SUBW2,3					
SUBL2,3					
DECB	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	1
DECW					
DECL					
SBWC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	1
MULB2,3	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
MULW2,3					
MULL2,3	N	Z	WMUX[L] ≠Ø	C	2
EMUL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
	SIGN(SZ)	WMUX(SZ)=Ø *Z	Ø	C	2
DIVB2,3	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
DIVW2,3					
DIVL2,3					

CCC Table 2 (Cont)

Integer Arithmetic & Logical Instructions: (Cont)

					CC
					OF
INSTR	N	Z	V	C	OP
EDIV	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
ASHL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
ASHQ	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
	SIGN(SZ)	WMUX(SZ)=Ø *Z	Ø	C	2
BITB BITW BITL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
BISB2,3 BISW2,3 BISL2,3	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
BICB2,3 BICW2,3 BICL2,3	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
XORB2,3 XORW2,3 XORL2,3	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
ROTL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2

CCC Table 3

Floating Point Instructions:

INSTR	N	Z	V	C	CC OF OP
MOVF MOVD	WBUS<15>	WMUX(SZ)=Ø	Ø	C	2
CLRF	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2
CLRD	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	1
MNEGF MNEGD	WBUS<15>	WMUX(SZ)=Ø	Ø	Ø	1
CVTFB	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
CVTDB	N	Z	WMUX[L] ≠Ø	C	2
CVTFW					
CVTDW					
CVTFL CVTDL CVTRFL CVTRDL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
CVTFD CVTDF CVTBF CVTBD CVTWF CVTWD CVTLD CVTLF	WBUS<15>	WMUX(SZ)=Ø	Ø	Ø	1
CMPF	WBUS<15>	WMUX(SZ)=Ø	Ø	Ø	1
	FP CMP N	WMUX(SZ)=Ø	Ø	Ø	2
TSTF TSTD ADDF2,3 ADDD2,3 SUBF2,3 SUBD2,3	WBUS<15>	WMUX(SZ)=Ø	Ø	Ø	1

CCC Table 3

Floating Point Instructions: (Continued)

ULF2,3					
MULD2,3					
DIVF2,3					
DIVD2,3					
EMODF					
EMODD					
POLYF					
POLYD					
CMPD	FP CMP N	WMUX(SZ)=Ø	Ø	Ø	2
	$\overline{\text{CRY}}$(SZ)	WMUX(SZ)=Ø	Ø	Ø	1

CCC Table 4

Address, Field, Control Instructions:

INSTR	N	Z	V	C	CC OF OP
MOVAB MOVAW MOVAL	SIGN (SZ)	WMUX (SZ) =Ø	Ø	C	2
MOVAQ	SIGN (SZ)	WMUX (SZ) =Ø	Ø	C	1
PUSHAB PUSHAW PUSHAL	SIGN (SZ)	WMUX (SZ) =Ø	Ø	C	2
PUSHAQ	SIGN (SZ)	WMUX (SZ) =Ø	Ø	C	1
FIELD INSTRUCTIONS					
EXTV EXTZV	SIGN (SZ)	WMUX (SZ) =Ø	Ø	C	2
INSV	SIGN (SZ)	WMUX (SZ) =Ø	OV (SZ)	C	2
CMPV	SIGN (SZ)				
	●				
CMPZV	OV (SZ)	WMUX (SZ) =Ø	Ø	$\overline{\text{CRY}}$ (SZ)	1
FFS	Ø	WMUX (SZ) =Ø	Ø	Ø	1
FFC	Ø	WMUX (SZ) =Ø	Ø	Ø	1
CONTROL INSTRUCTIONS					
ACBB ACBW	SIGN (SZ)	WMUX (SZ) =Ø	OV (SZ)	C	1
ACBL	SIGN (SZ)	WMUX (SZ) = Ø	OV (SZ)	C	2
ACBF ACBD	WBUS<15>	WMUX (SZ) =Ø	Ø	C	2
AOBLEQ AOBLSS SOBGEQ SOBGTR	SIGN (SZ)	WMUX (SZ) =Ø	OV (SZ)	C	2
CASEB	SIGN (SZ)	WMUX (SZ) =Ø	Ø	$\overline{\text{CRY}}$ (SZ)	1
	●				
CASEW CASEL	OV (SZ)				

CCC Table 5

Other Native Mode Instructions:

INSTR	N	Z	V	C	CC CT OP
MOVC3	Ø	WMUX(SZ)=Ø	Ø	Ø	2

MOVC5	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

MOVTC	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

MOVTUC	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

CMPC3
CMPC5
	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

SCANC SPANC	Ø	WMUX(SZ)=Ø	Ø	Ø	2
LOCC SKP MATCH	Ø	WMUX(SZ)=Ø	Ø	Ø	1
CRC	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2

MEMORY MANAGEMENT INSTRUCTIONS

PROBER PROBEW	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2

PROCESSOR REG INSTRUCTIONS

MTPR MFPR	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	2

CCC Table 5

Other Native Mode Instructions: (Cont)

					CC
					OF
INSTR	N	Z	V	C	OP
QUEUE INSTRUCTIONS					

INSQUE	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

REMQUE	SIGN(SZ)
	●
	OV(SZ)
		WMUX(SZ)=Ø	Ø	$\overline{\text{CRY}}$(SZ)	1

	N	Z	WBUS[L] =Ø	C	2
DECIMAL & NUMERIC STRING INSTRUCTIONS					
CVTLP	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
CVTPL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
ADAWI	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	2
INDEX	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	2

CCC Table 6

Compatibility Mode Instructions:

					CC
					OF
INSTR	N	Z	V	C	OP
MOVB MOV	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	1
CLRB CLR	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
NEGB NEG	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
COMB COM	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
CMPB CMP	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
TSTB TST	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1

CCC Table 6

Compatibility Mode Instructions: (Cont)

ADD	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
INCB INC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	C	1
ADCB ADC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
SUB	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
DECB DEC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	C	1
SBCB SBC	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	$\overline{\text{CRY}}$(SZ)	2
BITB BIT BISB BIS BICB BIC XOR	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	1
SXT	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	1

INSTR	N	Z	V	C	CC OF OP
MUL	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
	(SIGN(SZ) ● OV(SZ)) + CRY(SZ)	WMUX(SZ)=Ø *Z	Ø	WBUS[L] ≠Ø	2
DIV	SIGN(SZ)	WMUX(SZ)=Ø	OV(SZ)	CRY(SZ)	1
SWAB	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
RORB ROR	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31>	1
ROLB	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31:8> ≠Ø	2

CCC Table 6

Compatibility Mode Instructions: (Cont)

ROL	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31:16> ≠Ø	2
ASRB ASR	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31>	1
ASLB	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31:8> ≠Ø	2
ASL	SIGN(SZ)	WMUX(SZ)=Ø	N INPUT ● C INPUT	WBUS<31:16> ≠Ø	2
MFPI MFPD	SIGN(SZ)	WMUX(SZ)=Ø	Ø	C	1
MTPI MTPD					
ASH	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
ASHC	SIGN(SZ)	WMUX(SZ)=Ø	Ø	Ø	1
	SIGN(SZ)	WMUX(SZ)=Ø *Z	Ø	C	2

```
                   --------------
    D SIZE 0 H --->!01          48!<--> FPA PRESENT L
    D SIZE 1 H --->!02          47!<--- WBUS 15 H
     ALUZ B0 H --->!03          46!<--- ALUV 31 H
        IR 6 H --->!04          45!<--- ALUV 07 H
        IR 4 H --->!05          44!<--- ALUV 15 H
        IR 7 H --->!06          43!<--- ALUZ B1 H
        IR 5 H --->!07          42!<--- ALUC 31 H
        IR 3 H --->!08          41!<--- ALUC 15 H
        IR 2 H --->!09          40!<--- ALUC 07 H
        IR 1 H --->!10 ....... 39!<--- WBUS 31 H
        IR 0 H --->!11 .     . 38!---- GROUND
           VGA ----!12 . LID . 37!<--- ALUZ B3 H
           VCC ----!13 . DOWN. 36!<--> WBUS 07 H
       FPA 2 L --->!14 .     . 35!---- GROUND
  ARITH TRAP L <-->!15 ....... 34!<--> WBUS 06 H
       FPA V L --->!16          33!---> CCBR 1 H
       PSL C H <---!17          32!<--> WBUS 05 H
      CCBR 0 H <---!18          31!<--- ALUZ B2 H
   PROC INIT L --->!19          30!<--- WBUS 04 H
       B CLK L --->!20          29!<--- CC CTRL 3 H
     WBUS 00 H <-->!21          28!<--- CC CTRL 1 H
     WBUS 03 H <-->!22          27!<--- CC CTRL 2 H
     WBUS 01 H <-->!23          26!<--- CC CTRL 0 H
     WBUS 02 H <-->!24          25!<--- D CLK ENABLE H
                   --------------
```

CCC FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.8 CARRY LOOK AHEAD (CLA-DC612)

1. GENERAL DESCRIPTION:

This specification describes the detail requirements of a 40 pin configuration, Carry Look Ahead (CLA) device. The CLA chip is an array of combinational logic which uses propagate and generate signals to generate carries for up to eight ALU slices. The carries can be generated for either binary or BCD arithmetic as described herein. In addition to the carries into the ALU slices, the CLA generates four additional signals; a carry out of the entire ALU network (C8), a sign bit (FS), an overflow bit (FOV), and the "or" of all carries (C). These signals are also described below. The exact boolean equations for all signals generated in the CLA chip follow in CLA Table 2.

This chip also contains independent of the CLA logic a 2 to 1 inverting multiplexer with an open collector output.

CLA TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	G3B L		GA1TNF
2	C H		GA1TTN
3	C6 L		GA1TTI
4	P3 L		GA1TNF
5	G1B L		GA1TNF
6	C0 L		GA1TTI
7	SB H		GA1TNF
8	FS H		GA1TTN
9	FOV H		GA1TTN
10	BCDB L		GA1TZN
11	C0B L		GA1TNF
14	G7A L		GA1TNF
15	BINB L		GA1TZI
16	G5A L		GA1TNF
17	G3A L		GA1TNF
18	P2 L		GA1TNF
19	G6A L		GA1TNF
20	C7 L		GA1TTI
21	G2A L		GA1TNF
22	G6B L		GA1TNF
23	C5 L		GA1TTI
24	G4A L		GA1TNF
25	C3 L		GA1TTI
26	C1 L		GA1TTI
27	CI L		GA1TNF
28	G5B L		GA1TNF
29	G0A L		GA1TNF
30	P1 L		GA1TNF
31	G1A L		GA1TNF
32	P6 L		GA1TNF
33	P0 L		GA1TNF
34	G4B L		GA1TNF
36	G2B L		GA1TNF
37	C4 L		GA1TTI
39	P7 L		GA1TNF
40	C2 L		GA1TTI
41	P4 L		GA1TNF
42	PCD L		GA1TNF
43	P5 L		GA1TNF
44	G7B L		GA1TNF
45	MUX OUT L		GA1TTN
46	MUX IN B H		GA1TNF
47	MUX SEL H		GA1TNF
48	MUX IN A H		GA1TNF

2 **Performance:**

2.1 Binary Arithmetic: When the CLA chip is in the binary arithmetic mode (i.e. BCD L is at the high voltage), the carries are generated in the conventional manner for look-ahead. That is, the carry into a slice is a function of the carry in (C_I) and lower order Propagates (P's) and Generates (G's) according to the equation:

$$C_n = [(C_I)(P_0)(P_1)\dots(P_{n-1})] + [(G_0)(P_1)(P_2)\dots(P_{n-1})] + \dots + [(G_{n-2})(P_{n-1})] + G_{n-1}$$

2.2 BCD Arithmetic: When the CLA chip is in the BCD Arithmetic Mode (i.e. BCD L is at the low voltage), the order of significance of the P's, G's and C's is altered. The order is listed below in order of decreasing significance.

Most Significant: P_1, G_1, C_1
P_0, G_0, C_0
P_3, G_3, C_3
P_2, G_2, C_2
P_5, G_5, C_5
P_4, G_4, C_4
P_7, G_7, C_7
Least Significant: P_6, G_6, C_6

NOTE: This is shown more clearly in the equations depicted in CLA Table 2.

2.3 Other Signals:

C8: The CLA chip generates the carry out of the entire ALU network in a manner similar to the carries into the slices. This carry is also generated in one of two ways, depending on what mode the CLA chip is in.

Sign (FS): The CLA chip is used in the Floating Point Accelerator in the fraction data path. The floating point fractions are unsigned, therefore an extra bit beyond the data path is needed to indicate the sign of the sum or difference being calculated. This function has been designed into the CLA chip. C8 and the sign bit input (SB H) are used to generate the floating point fraction sign (FS H).

Overflow: In addition to a sign bit, the FPA needs an overflow indicator. This is provided by the overflow output of the CLA chip (FOV H).

C: The CLA chip generates the logical "or" of all carries (C H).

MUX IN A H, MUX IN B H, MUX SEL H, MUX OUT L: These are the signals associated with the multiplexor, and are defined by the following truth table.

MUX SEL H	MUX IN A H	MUX IN B H	MUX OUT L
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$\begin{aligned}C0 = {} & \overline{(BCD)}C_I + (BCD)(G_3 + P_3G_2 + P_3P_2G_5 + P_3P_2P_5G_4 + P_3P_2P_5P_4G_7 + \\ & P_3P_2P_5P_4P_7G_6 + P_3P_2P_5P_4P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C1 = {} & \overline{(BCD)}(C_IP_0 + G_0) + (BCD)(G_0 + P_0G_3 + P_0P_3G_2 + P_0P_3P_2G_5 + \\ & P_0P_3P_2P_5G_4 + P_0P_3P_2P_5P_4G_7 + P_0P_3P_2P_5P_4P_7G_6 + P_0P_3P_2P_5P_4P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C2 = {} & \overline{(BCD)}(C_IP_0P_1 + G_0P_1 + G_1) + (BCD)(G_5 + P_5G_4 + P_5P_4G_7 + \\ & P_5P_4P_7G_6 + P_5P_4P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C3 = {} & \overline{(BCD)}(C_IP_0P_1P_2 + G_0P_1P_2 + G_1P_2 + G_2) + (BCD)(G_2 + P_2G_5 + \\ & P_2P_5G_4 + P_2P_5P_4G_7 + P_2P_5P_4P_7G_6 + P_2P_5P_4P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C4 = {} & \overline{(BCD)}(C_IP_0P_1P_2P_3 + G_0P_1P_2P_3 + G_1P_2P_3 + G_2P_3 + G_3) + (BCD)(G_7 + \\ & P_7G_6 + P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C5 = {} & \overline{(BCD)}(C_IP_0P_1P_2P_3P_4 + G_0P_1P_2P_3P_4 + G_1P_2P_3P_4 + G_2P_3P_4 + G_3P_4 + \\ & G_4) + (BCD)(G_4 + P_4G_7 + P_4P_7G_6 + P_4P_7P_6C_I)\end{aligned}$$

$$\begin{aligned}C6 = {} & \overline{(BCD)}(C_IP_0P_1P_2P_3P_4P_5 + G_0P_1P_2P_3P_4P_5 + G_1P_2P_3P_4P_5 + G_2P_3P_4P_5 + \\ & G_3P_4P_5 + G_4P_5 + G_5) + (BCD)(C_I)\end{aligned}$$

$$\begin{aligned}C7 = {} & \overline{(BCD)}(C_IP_0P_1P_2P_3P_4P_5P_6 + G_0P_1P_2P_3P_4P_5P_6 + G_1P_2P_3P_4P_5P_6 + \\ & G_2P_3P_4P_5P_6 + G_3P_4P_5P_6 + G_4P_5P_6 + G_5P_6 + G_6) + \\ & \overline{(BCD)}(G_6 + P_6C_I)\end{aligned}$$

$$\begin{aligned}BINC8 = {} & \overline{(BCD)}(C_IP_0P_1P_2P_3P_4P_5P_6P_7 + G_0P_1P_2P_3P_4P_5P_6P_7 + G_1P_2P_3P_4P_5P_6P_7 + \\ & G_2P_3P_4P_5P_6P_7 + G_3P_4P_5P_6P_7 + G_4P_5P_6P_7 + \\ & G_5P_6P_7 + G_6P_7 + G_7)\end{aligned}$$

$$\begin{aligned}BCDC8 = {} & \overline{(BCD)}(G_1 + P_1G_0 + P_iP_0G_3 + P_1P_0P_3G_2 + P_1P_0P_3P_2G_5 + \\ & P_1P_0P_3P_2P_5G_4 + P_1P_0P_3P_2P_5P_4G_7 + P_1P_0P_3P_2P_5P_4P_7G_6 + \\ & P_1P_0P_3P_2P_5P_4P_7P_6C_I)\end{aligned}$$

FS = (BINC8)(SB) + (BINC8)(SB)

FOV = (BINC8)(SB)

C = $C_{\emptyset} + C_1 + C_2 + C_3 + C_4 + C_5 + C_6 + C_7 + BINC_8 + BCDC_8$

NOTE: BCD = BCD L

Pn = Pn L

G_n = Gn L

SB = SB L

CLA TABLE 2 - LOGIC EQUATIONS

2.4 Pin Descriptions

Signal Pins	Function	Electrical Characteristics	Notes
CI L	IN		
PØ L	IN		
P1 L	IN		
P2 L	IN		
P3 L	IN		
P4 L	IN		
P5 L	IN		
P6 L	IN		
P7 L	IN		
GØA L	IN		SEE NOTE A
GØB L	IN		↑
G1A L	IN		|

Signal Pins	Function	Electrical Characteristics	Notes
G1B L	IN	│	
G2A L	IN	│	
G2B L	IN	│	
G3A L	IN	│	
G3B L	IN	│	
G4A L	IN	│	
G4B L	IN	│	
G5A L	IN	│	
G5B L	IN	│	
G6A L	IN	│	
G6B L	IN	│	
G7A L	IN	│	
G7B L	IN		SEE NOTE A
BCD L	IN		
SB H	IN		
C0 L	OUT	TOTEM POLE	
C1 L	OUT	↑	
C2 L	OUT	│	
C3 L	OUT	│	
C4 L	OUT	│	
C5 L	OUT	│	
C6 L	OUT	│	
C7 L	OUT	TOTEM POLE	
BINC8 L	OUT	TRI-STATE	SEE NOTE B
BCDC8 L	OUT	TRI-STATE	SEE NOTE B
C H	OUT	TOTEM POLE	
FS H	OUT	TOTEM POLE	
FOV H	OUT	TOTEM POLE	
MUX SEL H	IN		
MUX In A H	IN		
MUX In B H	IN		
MUX OUT L	OUT	Open Collector	

NOTES:

A: The G's into the CLA chip are brought into this chip through two pins each for internal buffering reasons. The A's and B's will be connected together externally making in effect one pin.

B: Because of internal gating restrictions, BIN C8 and BCD C8 are generated through two tri-state drivers, and muxed by enabling one or the other with BCD L. These pins will be connected externally creating in effect one C8 L pin.

```
                 ---------------
    G3B L --->!01                48!<--- MUX IN A H
      C H <---!02                47!<--- MUX SEL H
     C6 H <-->!03                46!<--- MUX IN B H
     PE L --->!04                45!---> MUX OUT L
    G1B L --->!05                44!<--- G7B L
     C0 L <-->!06                43!<--- P5 L
     SB H --->!07                42!<--- BCD L
     PS H <---!08                41!<--- P4 L
    POV H <---!09                40!<--> C2 L
   BCDB L <---!10 ....... 39!<--- P7 L
    G0B L --->!11 .       . 38!---- GROUND
      VGA ----!12 . LID . 37!<--> C4 L
      VCC ----!13 . DOWN. 36!<--- G2B L
    G7A L --->!14 .       . 35!---- GROUND
   BINB L <-->!15 ....... 34!<--- G4B L
    G5A L --->!16                33!<--- P0 L
    G3A L --->!17                32!<--- P6 L
     P2 L --->!18                31!<--- G1A L
    G6A L --->!19                30!<--- P1 L
     C7 L <-->!20                29!<--- G0A L
    G2A L --->!21                28!<--- G5B L
    G6B L --->!22                27!<--- CI L
     C5 L <-->!23                26!<--> C1 L
    G4A L --->!24                25!<--> C3 L
                 ---------------
```

CLA FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN CLARIFICATION

3.9 CACHE MEMORY CONTROL (CMK-DC623)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements of a Cache Memory Control Chip (CMK). The CMK interfaces to the control portion of the memory interconnect (CMI). It generates the necessary control signals to initiate memory cycles and returns status to the rest of the memory interface logic at the completion of each memory cycle.

CMK TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	B CLK L		GA1TNF
2	ADD REG ENA L		GA1TTN
3	CMI 27 H		GA1TZF
4	LATCHED BUS 3 H		GA1TNF
5	LATCHED BUS 1 H		GA1TNF
6	LATCHED BUS 2 H		GA1TNF
8	CMI 31 H		GA1TZN
9	BUS 4 H		GA1TNF
10	M CLK ENABLE H		GA1TNF
11	CMI 29 H		GA1TZN
14	LATCHED BUS 0 H		GA1TNF
15	CMI 28 H		GA1TZN
16	CMI 30 H		GA1TZN
17	MA 01 H		GA1TNF
18	MA 00 H		GA1TNF
19	CMI CPU PRIORITY L		GA1TNF
20	DST RMODE H		GA1TNF
21	D SIZE 0 H		GA1TNF
22	D SIZE 1 H		GA1TNF
23	MMUX SEL S1 H		GA1TNF
24	HOLD L		GA1TCF
25	SNAPSHOT CMI L		GA1TTF
26	MSEQ INIT L		GA1TNF
27	ENA CMI L		GA1TTN
28	DBBZ L		GA1TCF
29	INT GRANT H		GA1TNF
30	PHASE 1 H		GA1TNF
31	WAIT H		GA1TNF
32	HIT H		GA1TNF
33	CACHE INT L		GA1TNF
34	INHIBIT CMI H		GA1TNF
36	STATUS 1 H		GA1TTF
37	CMI 25 H		GA1TZF
39	STATUS VALID L		GA1TTN
40	ST 1 L		GA1TCF
41	GRANT STALL L		GA1TTN
42	ST 0 L		GA1TCF
43	STATUS 0 H		GA1TTF
44	WRITE VECT OCC L		GA1TTN
45	CORR DATA INT L		GA1TTF
46	PREFETCH L		GA1TNF
47	CMI 26 H		GA1TZF
48	D CLK ENABLE H		GA1TNF

2 PERFORMANCE:

CMK I/O PINS

INPUTS	#PINS
B CLK L	1
BUS 4 H	1
CMI CPU PRIORITY L	1
D CLK ENABLE H	1
D SIZE <1:0> H	2
DST RMODE H	1
HIT H 1	
CACHE INT L	1
INHIBIT CMI H	1
INT GRANT H	1
LATCHED BUS <3:0> H	4
M CLK ENABLE H	1
MAD <01:00> H	2
MMUX SEL S1 H	1
MSEQ INIT L	1
PHASE 1 H	1
PREFETCH L	1
WAIT H	1

BI-DIRECTIONAL LINES (TRI-STATE UNLESS OTHERWISE NOTED)

CMI <27:25> H	3
DBBZ L	1 (OPEN-COLL.)
HOLD L	1 (OPEN-COLL.)
ST <1:0> L	2 (OPEN-COLL.)

OUTPUTS (TOTEM POLE UNLESS OTHERWISE NOTED)

ADD REG ENA L	1
CMI <31:28> H	4 (TRI-STATE)
CORR DATA INT L	1
ENA CMI L	1
GRANT STALL L	1
SNAPSHOT CMI L	1
STATUS <1:0> H	2
STATUS VALID L	1
WRITE VECT OCC L	1

2.1 The following is a list of the code assignments for the BUS FUNCTION Micro field:

0. READ PHYSICAL ADDRESS
1. PROCESSOR INITIALIZE
2. READ, NO MICRO-TRAP
3. I/O INITIALIZE
4. READ LOCK TIMEOUT TEST
5. NOP
6. READ, SECOND REFERENCE
7. NOP
8. WRITE PHYSICAL ADDRESS
9. REI CHECK
A. WRITE, SECOND REFERENCE
B. WRITE UNLOCK, SECOND REFERENCE
C. WRITE, NO MICRO-TRAP
D. NOP
E. WRITE LONGWORD, NO MICRO-TRAP
F. BUS GRANT
10. READ
11. READ LONGWORD
12. PTE ACCESS CHECK, WRITE
13. READ LOCK
14. READ WITH MODIFY INTENT
15. READ LONGWORD WITH MODIFY INTENT
16. PTE ACCESS CHECK, READ
17. PTE ACCESS CHECK, READ, KERNEL MODE
18. WRITE
19. WRITE LONGWORD
1A. WRITE IF NOT RMODE
1B. WRITE UNLOCK
1C. PROBE ACCESS, WRITE, MODE SPECIFIED
1D. PROBE ACCESS, WRITE
1E. PROBE ACCESS, READ, MODE SPECIFIED
1F. PROBE ACCESS, READ

Bus Functions are decoded from "LATCHED BUS 4 H"(FLIP-FLOP) and "LATCHED BUS <3:0> H"

2.2 Chip Outputs:

>>> CMI<31:28>, when enabled, are functions of:

LATCHED BUS FUNCTION
PREFETCH CYCLE
LATCHED D-SIZE
MAD <01:00> H

CMI<31:28> drivers are enabled during "ENA FUNC H"

"LATCHED BUS 4" (FLIP-FLOP) and "LATCHED BUS <3:0>" are decoded and the decoded functions are enabled into latches during "ADD REG ENA H" to provide LATCHED BUS FUNCTIONS.

"PREFETCH H" is also enabled into a latch during "ADD REG ENA H" to provide "PREFETCH CYCLE H".

"D-SIZE <1:0> H" are enabled into latches during:

"ADD REG ENA H" & "PREFETCH L" & "LATCHED BUS 4 H" &
"LATCHED BUS 3 H"

to provide LATCHED D-SIZE.

A prefetch cycle causes all four bits to be ones. In addition, the following latched bus functions cause all four bits to be ones:

READ PHYSICAL ADDRESS
READ LOCK TIMEOUT TEST
READ, SECOND REFERENCE
BUS GRANT
WRITE LONG, NO MICRO-TRAP
WRITE PHYSICAL ADDRESS
WRITE LONG
READ LONG WITH MODIFY INTENT
READ LONG

In all other cases, CMI<31:28> are produced according to the following three tables:

For all READs:

MA<1:0>

00	01	10	11
1111	1110	1100	1000

For all WRITEs except WRITE, SECOND REFERENCE and WR!TE UNLOCK, SECOND REFERENCE:

	MA<1:0>			
LATCHED D-SIZE<1:0>	00	01	10	11
00	0001	0010	0100	1000
01	0011	0110	1100	1000
10	1111	1110	1100	1000
11	1111	1110	1100	1000

FOR WRITE, SECOND REFERENCE AND WRITE UNLOCK, SECOND REFERENCE:

	MA<1:0>			
LATCHED D-SIZE<1:0>	00	01	10	11
00	0001	0001	0001	0001
01	0001	0001	0001	0001
10	0001	0001	0011	0111
11	0001	0001	0011	0111

>>> When enabled, CMI<27:25> are functions of:

PREFETCH CYCLE
LATCHED BUS FUNCTIONS
LATCHED D-SIZE
MAD <01:00> H

The tri-state drivers for CMI<27:25> are also enabled during "ENA FUNC H".

If "PREFETCH CYCLE H" is TRUE, CMI<27:25> will be: 000 (ALL LOW).

Otherwise, CMI 27 H will be 1 (HIGH) for all WRITES.

CMI 26 H will be 1 (HIGH) for:

READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
READ LOCK TIMEOUT TEST

CMI 25 H will be 1 (HIGH) if "LOCK H" is TRUE.

"LOCK H" is TRUE if:

READ LOCK
WRITE UNLOCK, SECOND REFERENCE
WRITE UNLOCK & "SUPPRESS LOCK L"

"SUPPRESS LOCK H" is TRUE if:

"MAD 01 H" & "MAD 00 H" & LATCHED D-SIZE 0 H +
LATCHED D-SIZE 1 H & ("MAD 01 H" + "MAD 00 H")

>>> "DBBZ L" is an open-collector driver, and is driven low during "ENA FUNC H" .

>>> "HOLD L" is an open-collector driver, and is driven low if:

"SNAPSHOT CMI H" & "INVAL WRITE L"

>>> "ST <1:0> L" are open collector drivers, and are both driven low during "WRITE VECTOR H" .

>>> "ADD REG ENA L" is driven directly from the "ADD REG ENA H" flip-flop and is LOW when the flip-flop is set.

>>> "CORR DATA INT L" is driven directly from the "CORR DATA INT H" flip-flop and is LOW when the flip-flop is set.

>>> "ENA CMI L" is true (LOW) if:

"SET ENA CMI H" & "SET BUSY H"

"SET ENA CMI H" is true if:

"CMI REQUEST H" + CMI ENABLE Latch is set

CMI ENABLE Latch is enabled while "B CLK L" is LOW. Latch input is:

("CMI REQUEST H" & "SET BUSY H" & "CMI ENA L") +
("DBBZ H" & "READ L" & "CMI ENA H")

"CMI REQUEST H" is true if:

"HOLD L" & "DBBZ L" & "CMI ENA L" & "MSEQ INIT L" &
"CPU PRIORITY H" & "READ LOCK INHIBIT L"

"READ LOCK INHIBIT H" is true if:

"LOCK SET H" & READ LOCK(Latched bus function)

"READ H" is enabled into a latch during "ADD REG ENA H" and is true if:

"PREFETCH H" +
(READ LOCK TIMEOUT TEST + Any READ)Bus Function Decodes

"SET BUSY H" is true if:

"CMI ENA H" + LATCHED BUSY is set.

LATCHED BUSY Latch is enabled while "B CLK L" is LOW. Latch input is:

("ADD REG ENA H" & "INHIBIT BUSY L"
& "BUSY L") +
("CMI IN PROG L" &"MSEQ INIT L" &
"TIMEOUT L" & "BUSY H")

"INHIBIT BUSY H" is true if:

"PREFETCH L" & (BUS GRANT(Bus function decode) +
"D CLK ENABLE L" + "M CLK ENABLE L") +

"INHIBIT CMI H" +

"READ H" & "READ LOCK L" & "HIT H"

"READ LOCK H" is true if:

"PREFETCH L" & READ LOCK(Bus Function Decode)

>>> "GRANT STALL L" is true (LOW) if:

```
BUS GRANT(Bus function decode) &
[("PREFETCH L" & "BUSY H" & "MSEQ INIT L") +
("PHASE 1 L" & "SYN INT DONE L" &
"WRITE VECTOR L" & "MSEQ INIT L")]
```

>>> "SNAPSHOT CMI L" is driven directly from the "SNAPSHOT CMI H" flip-flop, and is LOW when the flop is set.

>>> "STATUS <1:Ø> H" are driven directly from the "STATUS 1 H" and "STATUS Ø H" flip-flops, and are HIGH when the respective flops are set.

>>> "STATUS VALID L" is LOW if the "STATUS VALID H" flip-flop is set, or if "B CLK H" and the D-Input of the "STATUS VALID H" flip-flop is true.

>>> "WRITE VECT OCC L" is driven directly from the "WRITE VECTOR H" flip flop, and is LOW when the flop is set.

2.3 Read Lock Timeout Counter: The TIMEOUT COUNTER is a synchronous counter which is clocked by "TIMEOUT CLK H". "TIMEOUT CLK H" is produced by gating "B CLK H" with the output of a toggle flop which changes state on each rising edge of "B CLK L" except when "MSEQ INIT H" is true (HIGH). During "MSEQ INIT H" the toggle flop is steered such that it is clocked to the state which allows each "B CLK H" to produce a "TIMEOUT CLK H". The flip-flops in the TIMEOUT COUNTER are clocked on the leading (rising) edge of "TIMEOUT CLK H".

"TIMEOUT H" is true if all 8 bits of the TIMEOUT COUNTER are set (ONES) and the flop which gates "TIMEOUT CLK H" is reset ("TIMEOUT CLK H" disabled).

The TIMEOUT COUNTER is enabled if:

```
("LOCK SET H" & "BUSY H") +
READ LOCK TIMEOUT TEST(Bus Function Decode)
```

The TIMEOUT COUNTER has two counting modes. If "LOCK H" is true, the counter operates as a synchronous binary counter.

If "LOCK H" is false, all bits of the counter are inhibited from resetting unless the counter is full. Bits are set, on the rising edge of "TIMEOUT CLK H", when all less significant bits are set.

In either mode, the counter counts to all zeros on the clock after it counts to all ones.

The counter is cleared on the rising edge of "TIMEOUT CLK H" when the least significant bit is zero and the counter enable terms are false.

FLIP-FLOPS clocked on falling edge of "B CLK L":

#"SNAPSHOT CMI H" - JK FLIP-FLOP J Input:

"ADD CYC H" & "CMI 27 H" & "CMI 26 L" & "CMI IN PROG L"

"ADD CYC H" is true if:

"DBBZ H" & "DBBZ DEL L" & "MSEQ INIT L"

K Input:

"INVAL WRITE H" + "MSEQ INIT H"

#"LOCK SET H" - JK FLIP-FLOP J Input:

"ADD CYC H" & "CMI 27 L" & "CMI 26 L" & "CMI 25 H" &
"CMI IN PROG L"

K Input:

("ADD CYC H" & "CMI 27 H" & "CMI 26 L" & "CMI 25 H") +
"MSEQ INIT H" + "TIMEOUT H"

#"STATUS 1 H" - D FLIP-FLOP D Input:

"ST 1 H" + "FINAL CYC DEC L"

Clocking of the flop is inhibited if "STATUS VALID H" is true.

The flop is DC SET when "MSEQ INIT H" is TRUE.

The flop is DC cleared if:

"TIMEOUT H" +

"CACHE INT H" & ("STATUS VALID H" + "B CLK L")

"FINAL CYC DEC H" is TRUE if:

"CMI IN PROG H" & "DBBZ L"

#"STATUS 0 H" - D FLIP-FLOP D Input:

"ST 0 H" + "FINAL CYC DEC L"

Clocking of the flop is inhibited if "STATUS VALID H" is true.

The flop is DC cleared when "TIMEOUT H" is TRUE.

FLIP-FLOPS clocked on rising edge of "B CLK L":

#"ENA FUNC H" - D FLIP-FLOP D Input:

"SET BUSY H" & "SET ENA CMI H" & "DBBZ L"

#"ADD REG ENA H" - JK FLIP-FLOP J Input:

"MEM REQ H" & "INVAL CHECK L"

"MEM REQ H" is TRUE if:

["PREFETCH DEL H" +
("BUS CYC DEC H" & "PREFETCH L" & "REPLACEMENT L")] &
"BUSY L" & "CMI IN PROG L"

"REPLACEMENT H" is TRUE if:

"STATUS VALID H" & "ADD ENA DEL L" & "READ H"

#"ADD ENA DEL H" - D FLIP-FLIP D INPUT:

"ADD REG ENA H"

"BUS CYC DEC H" is TRUE for: (Bus function decodes)

READ PHYSICAL ADDRESS
READ, NO MICRO-TRAP
READ LOCK TIMEOUT TEST
READ SECOND REFERENCE
WRITE PHYSICAL ADDRESS
WRITE, SECOND REFERENCE
WRITE UNLOCK, SECOND REFERENCE
WRITE, NO MICRO-TRAP
WRITE LONGWORD, NO MICRO-TRAP
BUS GRANT
READ
READ LONGWORD
READ LOCK
READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
WRITE
WRITE LONGWORD
WRITE UNLOCK
"DST RMODE L" & WRITE IF NOT RMODE

K Input:

"PREFETCH CYCLE H" + "M CLK ENABLE H"

#"PREFETCH DEL H" - D FLIP-FLOP
D Input:

"PREFETCH H"

The flop is DC CLEARED if "PREFETCH H" is FALSE.

#"LATCHED BUS 4 H" - D FLIP-FLOP

D Input:

("BUS 4 H" & "M CLK ENABLE H") +
("LATCHED BUS 4 H" & "M CLK ENABLE L")

#"INVAL CHECK H" - JK FLIP-FLOP
J Input:

"ENABLE INVAL H" & "SNAPSHOT CMI H"

"ENABLE INVAL H" is TRUE if:

"MMUX SEL S1 L" &
("MEM REQ L" + "ADD REG ENA H" & "RESET ADD ENA H")

"RESET ADD ENA H" is TRUE if:

"PREFETCH CYCLE H" + "M CLK ENABLE H"

K Input:

"SNAPSHOT CMI L"

#"INVAL WRITE H" - D FLIP-FLOP
D Input:

"INVAL CHECK H"

The flop is DC cleared when "INVAL CHECK H" is FALSE.

#"WRITE VECTOR H" D FLIP-FLOP
D Input:

"TIMEOUT H" + WR VECT LATCH is set

WR VECT LATCH Input:

"ADD CYC H" & "CMI 27 H" & "CMI 26 H" & "CMI 25 L"

The latch is enabled while "B CLK L" is HIGH.

#"CORR DATA INT H" - JK FLIP-FLOP
J Input:

"STATUS 1 H" & "STATUS 0 L" &
"STATUS VALID H"

K Input:

"M CLK ENABLE H"

#"STATUS VALID H" - D FLIP-FLOP
D Input:

[("INHIBIT BUSY H" + "ADD REG ENA L") &
("ADD REG ENA H" & "RESET ADD ENA H +
"FINAL CYC DEC H" + "TIMEOUT H")] +
"MSEQ INIT H"

#"BUSY H" - JK FLIP-FLOP
J Input:

"INHIBIT BUSY L" & "ADD REG ENA H"

K Input:

"CMI IN PROG H" + "MSEQ INIT H" + "TIMEOUT H"

#"CMI IN PROG H" - JK FLIP-FLOP
J Input:

"SET BUSY H" & "SET ENA CMI H"

K Input:

"DBBZ L" + "MSEQ INIT H"

#"DBBZ DEL H" - D FLIP-FLOP
D Input:

"DBBZ H" & "MSEQ INIT L"

#"SYN INT DONE H" - D FLIP-FLOP
D Input:

"WAIT L" & "INT GRANT L" & INT LATCH is set

INT LATCH is set by "INT GRANT H" and reset by "PHASE 1 H".

#"CMI ENA H" - JK FLIP-FLOP
J Input:

"CMI REQUEST H" & "SET BUSY H"

K Input:

"READ H" + "DBBZ L" + "MSEQ INIT H"

```
                               ---------------
             B CLK L --->!01             48!<--- D CLK ENABLE H
      ADD REG ENA L <---!02              47!<--> CMI 26 H
           CMI 27 H <-->!03              46!<--- PREFETCH L
     LATCHED BUS 3 H --->!04             45!---> CORR DATA INT L
     LATCHED BUS 1 H --->!05             44!---> WRITE VECT OCC L
     LATCHED BUS 2 H --->!06             43!---> STATUS 0 H
                         !07             42!<--> ST 0 L
           CMI 31 H <-->!08              41!---> GRANT STALL L
            BUS 4 H --->!09              40!<--> ST 1 L
      M CLK ENABLE H --->!10 ....... 39!---> STATUS VALID L
           CMI 29 H <-->!11 .       . 38!---- GROUND
                 VGA ----!12 . LID . 37!<--> CMI 25 H
                 VCC ----!13 . DOWN. 36!---> STATUS 1 H
     LATCHED BUS 0 H --->!14 .       . 35!---- GROUND
           CMI 28 H <-->!15 ....... 34!<--- INHIBIT CMI H
           CMI 30 H <-->!16              33!<--- CACHE INT L
            MA 01 H --->!17              32!<--- HIT H
            MA 00 H --->!18              31!<--- WAIT H
  CMI CPU PRIORITY L --->!19             30!<--- PHASE 1 H
         DST RMODE H --->!20             29!<--- INT GRANT H
          D SIZE 0 H --->!21             28!<--> DBBZ L
          D SIZE 1 H --->!22             27!---> ENA CMI L
       MMUX SEL S1 H --->!23             26!<--- MSEQ INIT L
              HOLD L <-->!24             25!---> SNAPSHOT CMI L
                               ---------------
```

CMK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.10 CONSOLE INTERFACE CHIP (CON-DC611)

1. GENERAL DESCRIPTION:

The console interface chip (CON) is an asynchronous serial line interface. It contains logic to do limited character recognition of the received characters for entering in Console Mode. It asserts signals to request interrupts. Addressing of internal registers is indirect through Console Register Address Register (CRAR).

CON Table 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	WCTRL 0 H		GA1TNF
2	MSEQ INIT L		GA1TNF
3	WCTRL 5 H		GA1TNH
4	WCTRL 2 H		GA1TNF
5	CHIP SEL L		GA1TNH
6	HALT DET BR H		GA1TTN
7	DONE BR L		GA1TTF
8	FRNT PNL LOCK H		GA1TNF
9	SERIAL IN H		GA1TNF

COM Table 1 (Cont)

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1Ø	CLR DONE IN L		GA1TNF
11	T READY SYNC L		GA1TNF
14	RD INT INH H		GA1TNF
15	WBUS ØØ H		GA1TZF
16	CON D CLK L		GA1TTN
17	INIT TT CTR H		GA1TNF
18	BAUD RATE CLK H		GA1TNF
19	WBUS Ø3 H		GA1TZF
2Ø	WBUS Ø1 H		GA1TZF
21	WBUS Ø2 H		GA1TZF
22	WBUS Ø5 H		GA1TZF
23	WBUS Ø4 H		GA1TZF
24	WBUS Ø7 H		GA1TZF
25	SERIAL OUT H		GA1TTN
26	HALT LIGHT L		GA1TCN
27	CON HALT L		GA1TTF
28	INSTR FETCH H		GA1TNF
29	LOAD CLK H		GA1TTN
3Ø	WBUS Ø6 H		GA1TZF
31	M CLK L		GA1TNF
32	WCTRL 1 H		GA1TNF
33	HALT DET SYNC H		GA1TNH
34	CLR CTDBE IN L		GA1TNF
36	WCTRL 3 H		GA1TNF
37	D CLK ENABLE H		GA1TNF
39	WBUS Ø8 H		GA1TZN
4Ø	WBUS Ø9 H		GA1TZN
41	CON INT L		GA1TPF
42	M CLK L		GA1TNF
43	CLR CTDBE OUT L		GA1TTF
44	T READY BR L		GA1TTN
45	FRM ERR EN L		GA1TNF
46	WCTRL 4 H		GA1TNF
47	CLR DONE OUT L		GA1TTF
48	DONE SYNC L		GA1TNF

2 Performance Requirements:

2.1 WBUS <7:0> H: WBUS <7:0> H is an eight bit tri-state transceiver. It is a port by which the internal registers are accessed. If READ CRAR, WBUS <9:8> H drivers are enabled, and are driven with CRAR <1:0> respectively. If READ (CRAR), WBUS <7:0> H drivers are enabled. Data is selected as in CON Table 2.

CON Table 2

DATA SELECTION

WBUS	CRAR = 0	CRAR = 1	CRAR = 2	CRAR = 3
7	CRDB <7:0> H	T READY H * RD INT INH L	DONE H	HALT PEND H
6	CRDB <7:0> H	CTCS IE H	CRCS IE H	HALT H
5	CRDB <7:0> H	0	0	T INT H * RD INT INH L
4	CRDB <7:0> H	0	0	0
3	CRDB <7:0> H	0	0	0
2	CRDB <7:0> H	0	0	0
1	CRDB <7:0> H	0	0	0
0	CRDB <7:0> H	0	0	0

2.2 CLOCKS: CON inputs 3 clocks, and outputs one.

A. Input - M CLK L - Note: M CLK L is received on two signal pins.

B. Input - D CLK ENABLE H

C. Input - BAUD RATE CLK H - This signal is called BR CLK H in the spec for simplicity.

D. Output - CON D CLK L:

D CLK L = M CLK L + D CLK ENABLE L
CON D CLK L = D CLK L

2.3 MSEQ INIT L: MSEQ INIT L is a TTL input used for initialization.

2.4 REGISTER ACCESS: There are five main registers in the console interface chip. They are summarized in COM Table 3. The Console Register Address Register (CRAR) is a two bit register used to address the internal registers. It is encoded as shown in the Register Address column in COM Table 3.

COM Table 3

REGISTER DATA

REGISTER	REGISTER ADDRESS	READ/WRITE
CTDB	00	W
CTCSR	01	R/W
CRDB	00	R
CRCSR	10	R/W
CSR	11	R/W

Register Access (Cont.): Four WCTRL functions are required to operate the console chip:

1. WRITE CRAR
2. WRITE (CRAR)
3. READ (CRAR)
4. READ CRAR

The CRAR is a two bit transparent latch, loaded from WBUS<7:6> with the low pulse of D CLK L if WRITE CRAR. It is read to WBUS <9:8> if READ CRAR.

WCTRL <5:0> H, CHIP SEL L: WCTRL <5:0> H and CHIP SEL L are TTL inputs used for control. A valid WCTRL function code is detected as follows:

VALID FUN = WCTRL 5 L .AND. WCTRL 4 H .AND. WCTRL 1 H .AND.
WCTRL 0 H .AND. CHIP SEL H

The chip functions are decoded as shown below:

VALID FUN	WCTRL 3 H	WCTRL 2 H	Function
0	X	X	NOP
1	0	0	WRITE CRAR
1	0	1	WRITE (CRAR)
1	1	0	READ CRAR
1	1	1	READ (CRAR)

2.5 Transmitter: The transmitter is a double buffered parallel to serial interface.

Transmitter Registers: There are two addressable registers in the transmitter.

1. Conscle Transmitter Buffer Register (CTBR). This is an 8 bit data register loaded from WBUS<7:0>. It is write only under control cf WCTRL <5:0> H and the CRAR. The register is loaded with the low pulse of D CLK L.

2. Console Transmitter Control/Status Register (CTCSR). This is a two bit register accessed from the WBUS under control of WCTRL <5:0> H and the CRAR. It contains an Interrupt Enable (IE) bit (CTCSR<6>) and a READY bit (CTCSR<7>). IE is READ/WRITE, and is loaded from WBUS<6> with the low pulse of D CLK L. READY is read only to WBUS<7>.

Transmitter Operation: The transmitter accepts 8-bit data from WBUS <7:0>. If LOAD (CRAR), and the CRAR points to data buffer, the data is strobed from the WBUS into the CTDB with D CLK L. This clears the READY flip-flop. The register from which the data is actually transmitted is called the Transmitter Register (TR). If the Transmitter Register is empty, the data is automatically transfered from the CTDB into the TR, and READY is set. If the Interrupt Enable bit in the CTCSR is set, CON INT H gets asserted. This causes an interrupt to signal that the CTDB is ready to accept another character. This interrupt is cleared by writing a one to CSR<5>. When the TR is loaded, a counter is started which controls shifting the data out onto the Serial Out (SO) line. A one bit space is shifted out first (start bit), then the 8 data bits and then one mark (stop bit). The shift rate is determined by the frequency of the BR CLK H.

Transmitter Logic: The Transmitter Logic is described in the following sections.

1. TT CTR <3:0> H - INIT TT CTR H

   A. TT CTR <3:0> H is a four bit ripple counter clocked with the rising edge of BR CLK L.

   B. TT CTR <3:0> H is asynchronously forced to 0 if INIT TT CTR H = 1.

2. T CLK L, LD CLK L, LOAD CLK H

A. T CLK L is an internal clock signal. It is low when TT CTR <3:0> H = F and BR CLK L is low.

B. LD CLK L is an internal clock signal. It is low when TT CTR <3:0> H = 7 and BR CLK L is low.

C. LOAD CLK H is a TTL output. LOAD CLK H = .NOT. LD CLK L.

3. TC CTR <3:0> H: TC CTR <3:0> H is a four bit ripple counter clocked with the falling edge of T CLK L. It is asynchronously set to E when INIT TC CTR L is low.

4. CLR CTDBE IN L - CLR CTDBE DLY H:

A. CLR CTDBE IN L is the low true input to an edge triggered D flop, CLR CTDBE DLY H.

B. CLR CTDBE DLY H is clocked with the rising edge of LD CLK H. It is asynchronously cleared if MSEQ INIT H.

5. T READY BR H - T READY BR L:

A. T READY BR H is an edge triggered D Flop clocked with the rising edge of LD CLK H. It is cleared if CLR CTDBE DLY H * (TC CTR <3:0> H = E). It is only set asynchronously.
SET T READY BR H = MSEQ INIT H + (TC CTR <3:0> H = F) * T CLK L.

B. T READY BR L is a TTL output, and equals .NOT. T READY BR H.

6. INIT TC CTR H: INIT TC CTR H is an S-C latch.

A. SET INIT TC CTR H = (T CLK L * (TC CTR <3:0> H = 8))
+ MSEQ INIT H

B. CLR INIT TC CTR H = T CLK L * (TC CTR <3:0> H = E)
* LD CLK L * T READY BR L

7. CLR CTDBE OUT H - CLR CTDBE OUT L

A. CLR CTDBE OUT H is a S-C latch built with a TTL transceiver. SET CLR CTDBE OUT H = LD CTDB H
CLR CLR CTDBE OUT H = MSEQ INIT H + T READY BR L

B. The low true output is available at the output pin, CLR CTDBE OUT L.

8. CTDB <7:0> H: CTDB <7:0> H is an eight bit transparent latch. It is loaded from WBUS <7:0> H respectively. The latch is open if LD CTDB H, where LD CTDB H = WRITE (CRAR) * (CRAR = 0) * D CLK H

9. TR <7:0> H: TR <7:0> H is an eight bit transparent latch. It is loaded from CTDB <7:0> H respectively. The latch is open if INIT TC CTR H * T READY BR H * LD CLK H

10. XMIT SERIAL OUT H: XMIT SERIAL OUT H is a transparent latch, open if T CLK L is high. Data is selected as below:

    A. If TC CTR <3:0> H = F * INIT TC CTR L,
       then XMIT SERIAL OUT H <- 1

    B. If TC CTR <3> L,
       then XMIT SERIAL OUT H <-
       TR <(TC CTR <2:0> H)> L

    C. else XMIT SERIAL OUT H <- 0

11. SERIAL OUT H: SERIAL OUT H is a TTL output.
    SERIAL OUT H = .NOT. XMIT SERIAL OUT H

T READY SYNC L: T READY SYNC L is a TTL input.

T READY H: T READY H is a transparent latch, open when D CLK L is high. The data input is T READY SYNC H. It is asynchronously cleared if CLR CTDBE OUT H.

CTCS IE H: CTCS IE H is a transparent latch, open if D CLK H * WRITE (CRAR) * (CRAR = 2). The data input is WBUS 06 H. It is asynchronously cleared if MSEQ INIT H.

T INT H: T INT H is a flipflop set with the rising edge of CTCS IE H * T READY H if HALT L. It is asynchronously cleared if D CLK H * WRITE (CRAR) * (CRAR = 3) * WBUS 05 H.

2.6 Receiver: The Receiver is a double buffered serial to parallel interface.

Receiver Registers: There are two addressable registers in the receiver:

1. Console Receiver Data Buffer Register (CRDB) - this is an o bit read only register accessed from the WBUS. It contains data received on the SERIAL IN (SI) line.

2. Console Receiver Control/Status Register (CRCSR) - this is a two bit register accessed from the WBUS. It contains an Interrupt Enable bit and a DONE bit. IE is READ/WRITE, and is loaded from WBUS<6> with the low pulse of D CLK L. DONE is read only to WBUS<7>.

Receiver Operation: If the receiver is in the idle state, when the SERIAL LINE (SI) goes from a mark to a space, a divide by 16 Receiver Timing Counter (RT CTR) is started. It is incremented at the frequency of the baud rate CLK H signal. The SI is tested to verify a valid start bit (space). If not, the RT CTR is re-initialized. If yes, a second 4 bit counter, Receiver Control Counter (RC CTR), is enabled. This counter is incremented with the negative going edge of BR CLK H each time the RT CTR is in a state of 15.

The second counter is used to enable, in sequence, each of the eight latches in the Receiver Register. The latches are strobed at the center of the data pulse as determined by the state of the RC CTR. SI is tested for one stop bit (mark). If correct, the data is strobed into the CRDB, and the DONE bit in the CRCSR is set. If enabled, (i.e., the Interrupt Enable bit in the CRCSR is set) the setting of DONE will generally cause an interrupt. This interrupt is cleared by reading the CRDB. DONE is also cleared when the CRDB is read. The receiver is immediately ready to start accepting another character.

Receiver Logic: Receiver logic is described in the following sections.

RT CTR <3:0> H: RT CTR <3:0> H is a four bit ripple counter, clocked with the falling edge of BR CLK H unless R ERROR H + INIT RT CTR H. It is asynchronously cleared if INIT RT CTR H.

RC CTR <3:0> H: RC CTR <3:0> H is a four bit ripple counter, clocked with the falling edge of BR CLK H if RT CTR <3:0> = F.

SERIAL IN H: SERIAL IN H is a TTL input.

ST DET H is a transparent latch. It is open if BR CLK L * R BUSY L, data input is SERIAL IN L. It is asynchronously cleared if MSEQ INIT H + R BUSY H.

R BUSY H: R BUSY H is an S-C latch.
A. SET R BUSY H = ST DET H * BR CLK H
   * (RT CTR <3:0> H = 7)

B. CLR R BUS H = SERIAL IN H * R ERROR H + MSEQ INIT H + STOP BIT H * BR CLK L

R ERROR H: R ERROR H is an S-C latch.

A. SET R ERROR H = (RC CTR <3;)> H = 8) * (RT CTR <3:0> H = 7)
   * SERIAL IN L * BR CLK H

B. CLR R ERROR H = RBUSY L * ST DET L + MSEQ INIT H

STOP BIT H: STOP BIT H is an S-C latch. SET STOP BIT H = SERIAL IN H * (RC CTR <3:0> = 8) * (RT CTR <3:0> H = 7) * BR CLK H CLR STOP BIT H = R BUSY L + MSEQ INIT H

CLR DONE IN L: CLR DONE IN L is a TTL input.

CLR DONE DLY H: CLR DONE DLY H is an edge triggered D flip-flop clocked with the rising edge of BR CLK H. It is asynchronously set if MSEQ INIT H. The data input is CLR DONE IN H.

FRNT PNL LOCK H: FRNT PNL LOCK H is a TTL input.

DONE BR H - DONE BR L: DONE BR H is edge triggered to clear flipflop. Specifically, it is cleared with the rising edge of BR CLK H if CLR DONE DLY H. It is asynchronously cleared if MSEQ INIT H. It is asynchronously set if
STOP BIT H * BR CLK H * .NOT [HALT L * (RR <6:0> = 10 (HEX)) *
FRNT PNL LOCK L]. DONE BR L is a TTL output.

CLR DONE OUT H - CLR DONE OUT L: CLR DONE OUT H is an edge trigger to clear flipflop. Specifically, it is cleared with the rising edge of BR CLK H if CLR DONE DLY H. It is asynchronously set if MSEQ INIT H + READ (CRAR) * (CRAR = 0).CLR DONE OUT L is a TTL output.

FRM ERR EN L: FRM ERR EN L is a TTL input.

HALT DET BR H: HALT DET BR H is a TTL output. HALT DET BR H = FRM ERR EN H * R ERROR H + STOP BIT H * BR CLK H * FRNT PNL LOCK L * RR <6:0> = 10 (HEX).

RR <7:0> H: RR <7:0> H is an eight bit S-C latch. The latches are set one at a time.

A. If ST DET L * R BUSY L * .NOT> (STOP BIT H * BR CLK H * DONE BR L) then RR <7:0> <- 0.

B. If BR CLK H * (RT CTR <3:0> = 7) * RC CTR <3> L * SERIAL IN H then: RR <RC CTR <2:0> H> <- 1.

CRDB <7:0> H: CRDB <7:0> H is an eight bit transparent latch.

If STOP BIT H * BR CLK H * DONE BR L
then: CRDB <7:0> H <- RR <7:0> H

DONE SYNC L: DONE SYNC L is a TTL input.

DONE H: DONE H is a transparent latch, open when D CLK L is high. The data input is DONE SYNC H. It is asynchronously cleared if CLR DONE OUT H.

CRCS IE H: CRCS H is a transparent latch, open if D CLK H * WRITE (CRAR) * (CRAR = 1). The data input is WBUS 06 H. It is asynchronously cleared if MSEQ INIT H.

R INT H: R INT H is a flip-flop set with the rising edge of CRCS IE H * DONE H if HALT L. It is asynchronously cleared if MSEQ INIT H + D CLK H * READ (CRAR) * (CRAR = 0).

2.7 Interrupts & Halt Logic:

RD INT INH H: RD INT INH H is a TTL input.

CON INT H - CON INT L: CON INT H is a transparent latch, the output stage of which is an open collector transceiver. CON INT L is the low true output pin signal. The latch is open when M CLK L is high. The data input is R INT H + T INT H * RD INT INH L.

HALT DET SYNC H: HALT DET SYNC H is a TTL input.

HALT PEND H: HALT PEND H is an edge triggered D flop, clocked with the rising edge of D CLK L. If HALT DET SYNCH H, then HALT PEND H will be set. If WRITE (CRAR) * (CRAR = 3), then HALT PEND H <- WBUS 07 H + HALT DET SYNC H.

INSTR FETCH H: INSTR FETCH H is a TTL input.

SAVED INSTR FETCH H: SAVED INSTR FETCH H is an edge triggered D flop, clocked with the rising edge of M CLK L. The data input is INSTR FETCH H. It is asynchronously set if MSEQ INIT H.

HALT H: HALT H is an edge triggered D flop clocked with the rising edge of M CLK L. It cannot change state unless SAVED INSTR FETCH H + WRITE (CRAR) * (CRAR = 3).

The data input is [SAVED INSTR FETCH H * HALT PEND H + WRITE (CRAR) * (CRAR = 3) * WBUS 04 H] * .NOT. [WRITE (CRAR) 8 (CRAR = 3) * WBUS 06 H].
CON HALT L: CON HALT L is a TTL output, and equals HALT L.

HALT LIGHT L: HALT LIGHT L is an open collector output, and equals HALT L

```
                                  ---------------
          WCTRL 0 H --->!01              48!<--- DONE SYNC L
        MSEQ INIT L --->!02              47!<--> CLR DONE OUT L
          WCTRL 5 H --->!03              46!<--- WCTRL 4 H
          WCTRL 2 H --->!04              45!<--- FRM ERR EN L
         CHIP SEL L --->!05              44!---> T READY BR L
      HALT DET BR H <---!06              43!<--> CLR CTDBE OUT L
          DONE BR L <-->!07              42!<--- M CLK L
    FRNT PNL LOCK H --->!08              41!<--> CON INT L
        SERIAL IN H --->!09              40!---> WBUS 09 H
      CLR DONE IN L --->!10  ......      39!---> WBUS 08 H
   T READY SYNC L --->!11  .      .      38!---- GROUND
                VGA ----!12  . LID .     37!<--- D CLK ENABLE H
                VCC ----!13  . DOWN.     36!<--- WCTRL 3 H
       RD INT INH H --->!14  .      .    35!---- GROUND
         WBUS 00 H <-->!15  .......      34!<--- CLR CTDBE IN L
        CON D CLK L <---!16              33!<--- HALT DET SYNC H
      INIT TT CTR H --->!17              32!<--- WCTRL 1 H
    BAUD RATE CLK H --->!18              31!<--- M CLK L
          WBUS 03 H <-->!19              30!<--> WBUS 06 H
          WBUS 01 H <-->!20              29!---> LOAD CLK H
          WBUS 02 H <-->!21              28!<--- INSTR FETCH H
          WBUS 05 H <-->!22              27!<--> CON HALT L
          WBUS 04 H <-->!23              26!---> HALT LIGHT L
          WBUS 07 H <-->!24              25!---> SERIAL OUT H
                                  ---------------
```

CON FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.11 INTERRUPT CONTROL CHIP (INT-DC630)

1. GENERAL DESCRIPTION:

The INT chip performs various operations relevant to interrupt processing, specifically:

1. storing and returning values of parts of the PSL (IS,CURMODE,PRVMODE,IPL) and AST level via the WBUS

2. receiving and storing the value of the HSIPR (Highest pending Software Interrupt Priority level Request/Register), which is used in interrupt arbitration

3. placing various data onto the microvector (UVECTOR) lines

4. performing REI check calculations

5. arbitrating interrupt requests, encoding the identity of the highest priority request (for output onto the UVECTOR lines), and generating the interrupt pending signal, INT PNDG

6. arbitrating Unibus requests within the group of BR devices, and issuing BGs (thus granting interrupt access to the CPU, though access to the Unibus must be further arbitrated in other circuitry external to INT)

INT TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	WEI L		GA1TNF
2	CPI L		GA1TNF
3	SPFI L		GA1TNF
4	CDI L		GA1TNF
5	WCTRL 4 H		GA1TNF
6	WCTRL 5 H		GA1TNF
7	PHASE1 H		GA1TNF
8	WBUS 22 H		GA1TZF
9	WBUS 23 H		GA1TZF
10	PROC INIT L		GA1TNF
11	WCTRL 2 H		GA1TNF
14	WCTRL 1 H		GA1TNF
15	WCTRL 3 H		GA1TNF
16	WBUS 25 H		GA1TZF
17	WBUS 24 H		GA1TZF
18	WBUS 26 H		GA1TZF
19	WCTRL 0 H		GA1TNF
20	UVECTOR 0 H		GA1TZN
21	UVECTOR 2 H		GA1TZN
22	UVECTOR 1 H		GA1TZN
23	UTRAP L		GA1TNF
24	UVECTOR BRANCH H		GA1TNF
25	DOSRV L		GA1TNF
26	PTE CHK OR PROBE H		GA1TNF
27	WBUS 20 H		GA1TZF
28	WBUS 18 H		GA1TZF
29	DCLK EN H		GA1TNH
30	SLINE INT L		GA1TNF
31	WBUS 19 H		GA1TZF
32	WBUS 17 H		GA1TZF
33	WBUS 16 H		GA1TZF
34	MCLK EN H		GA1TNH
36	BCLK L		GA1TNF
37	SYNCHR RESET BG H		GA1TNG
39	SBR6 H		GA1TNF

INT TABLE 1 (Cont)

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
40	SBR7 H		GA1TNF
41	HPBG5 H		GA1TTN
42	SBR5 H		GA1TNF
43	HPBG4 H		GA1TTN
44	HPBG6 H		GA1TTN
45	SBR4 H		GA1TNF
46	BUS GRANT H		GA1TTN
47	INT PNDG L		GA1TTN
48	TIMER INT L		GA1TNF

2.0 **Performance:**

2.1 Introduction:

2.1.1 **Overview:** The INT chip performs various operations relevant to interrupt processing, specifically:

A. storing and returning values of parts of the PSL (IS,CURMODE,PRVMODE,IPL) and AST level via the WBUS

B. receiving and storing the value of the HSIPR (Highest pending Software Interrupt Priority level Request/Register), which is used in interrupt arbitration

C. placing various data onto the microvector (UVECTOR) lines

D. performing REI check calculations

E. arbitrating interrupt requests, encoding the identity of the highest priority request (for output onto the UVECTOR lines), and generating the interrupt pending signal, INT PNDG

F. arbitrating Unibus requests within the group of BR devices, and issuing BGs (thus granting interrupt access to the CPU, though access to the Unibus must be further arbitrated in other circuitry external to INT)

In the following discussions of INT operation, reference occassionally will be made to operation of other parts of the COMET CPU. The operator (INT or other) will hopefully be clear from the context.

2.1.2 **State Elements:** Various state elements are used in the INT chip, some level triggered, some edge triggered.

SR latches are the standard cross-coupled NAND gates with independent S and R inputs.

Transparent (or T) latches and data (or D) latches are three-gate level-triggered devices with two clock inputs and one data input. While clocks are deasserted, outputs (Q and notQ) hold their previous values. While clocks are asserted, the input D is fed through to the outputs. (Transparent latches have both outputs valid during clock assertion; D latches have the non-inverting output valid and the inverting output forced high.) Skew latches are like D or transparent latches with the clock inputs reversed. While clocks are asserted, the outputs may not change. When clocks are de-asserted, the outputs will follow the input, D. Skew latches are used to avoid a potential problem whereby a data input, intended to change on a clock trailing edge, may change a few nsec too soon because of clock skew. The skew latch effectively locks out this change until after the trailing edge of the locally available clock.

Edge-triggered FFs are master-slave devices whose outputs are held while the clock is de-asserted. While the clock is asserted, the FF sets up in response to the data input, but the outputs do not change until the trailing edge of the clock.

2.1.3 **Signal Names and Assertion Levels:** Throughout this document, the following logical operators will be used.

A. "+" or "OR" means "logical or".
B. "&" or "AND" means "logical and".
C. "not" means "logical inverse".

In the interest of minimizing confusion over truth values (T/F), logical values (1/0), and voltage levels (H/L), the following conventions will be used in this document.

A. A physical signal name will include a trailing H or L to indicate its voltage assertion level. Thus UTRAP L is asserted (T) when low, deasserted (F) when high.

B. H and L will be used only in physical signal names.

C. logical values (1/0) when applied to physical signal names will follow the voltage level such that 1 corresponds to H.

thus UTRAP L = 0 is asserted (T),
BUS GRANT H = 0 is deasserted (F).

D. Logical signal names (in logical equations) will not have H or L suffixes and will be asserted when 1.

Thus (UTRAP & notDOSRV) = 1 means UTRAP is asserted (logical UTRAP = 1 = T, physical UTRAP L = 0 = low = T) and DOSRV is not asserted (logical DOSRV = 0 = F, physical DOSRV L = 1 = high = F); i.e. the logical inverse of DOSRV, notDOSRV, is asserted = 1 = T.

2.2 **Input and Output Signals:**

2.2.1 **Chip I/O Pins:** I/O signals are listed by pin number in INT Table 1. They are shown here grouped according to function. Outputs are totem pole unless otherwise specified. See following Para. 2.2.2 for signal descriptions.

I/O	name	relevant section(s)
I	BCLK L	clocks
I	PHASE1 H	clocks
I	MCLK EN H	clocks
I	DCLK EN H	clocks
I	WCTRL <5:0> H	PSL, UVECTOR, REI, intpts, Bus Grant
I/O(3-st)		WBUS <26:22> H PSL, REI
I/O(3-st)		WBUS <20:16> H PSL, REI
I	DOSRV L	UVECTOR
I	UTRAP H	UVECTOR, intpts
I	PTE CHK OR PROBE H	UVECTOR, intpts
O(3-st)	UVECTOR <2:0> H	UVECTOR
I	SPFI L	intpts
I	WEI L	intpts
I	CPI L	intpts
I	CDI L	intpts
I	TIMER INT L	intpts
I	SLINE INT L	intpts
I	SBR7 H	intpts
I	SBR6 H	intpts
I	SBR5 H	intpts
I	SBR4 H	intpts
I	UVECTOR BRANCH H	intpts
I	PROC INIT L	intpts
O	INT PNDG L	intpt pending
I	SYNCHR RESET BG H	Bus Grant
O	BUS GRANT H	Bus Grant
O	HPBG6 H	Bus Grant
O	HPBG5 H	Bus Grant
O	HP3G4 H	Bus Grant

N.B. The above signal names are those used in the INT chip design and are similar to, but not necessarily identical to, the names assumed by other COMET chips or circuits.

2.2.2 Signal Descriptions:

Clocks: BCLK L is the basic system clock. Its period is T ns., duty cycle is approximately 35%. Most state elements (FFs) change on the trailing edge of BCLK. Transparent latches are enabled during BCLK. Skew latches are disabled during BCLK.

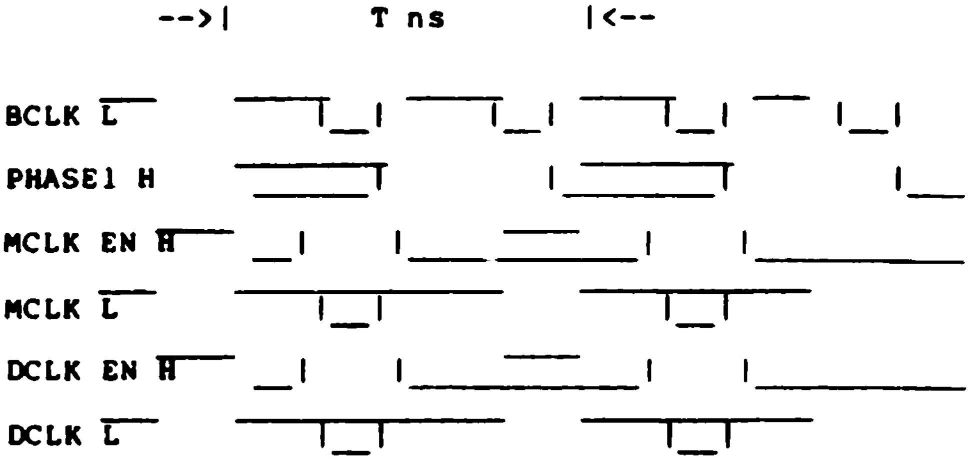
PHASE1 H is a square wave which changes state at the trailing edge of BCLK (+/- a few 10s of nsec because of propagation delays, etc.).

MCLK EN H is used to create MCLK (the microsequencer clock) from BCLK. MCLK normally (unless stalled -- see section) has twice the period (2T) and half the duty cycle of BCLK. An MCLK pulse occurs at the end of PHASE1 H deassertion. Therefore MCLK EN H (when present) spans the low to high transition of PHASE1 H.

DCLK EN H is used to create DCLK (the destination clock) from BCLK. DCLK timing relative to BCLK and PHASE1 is the same as that of MCLK. DCLK may also be stalled, during a "destination inhibit" operation.

Figure 1 shows the relative timing of these clock signals.

Data operations are typically performed at the trailing (rising) edge of DCLK L. Control operations typically occur at the trailing edge of MCLK L.



INT Figure 1 - Timing Diagram (Clock Signals)

WBUS/WCTRL: The WBUS is a tri-state data bus used (here) to write data relating to CPU status into and out of certain status registers on the INT chip. Ten of the WBUS lines are used by this chip.

WCTRL is a six-line control field. Those codes to which INT responds (commands to read/write status data from/to the WBUS, to issue a Unibus Grant, to place results of an REI check onto the UVECTOR lines, or to place certain sta+us data onto the UVECTOR lines) are enumerated in section 2.4.

Service Signals: INT responds to various other service/status signals as well.

DOSRV L signals hardware service conditions of such things as arithmetic traps, interrupts, etc.

UTRAP L indicates that the hardware is servicing a micro-trap condition.

PTE CHK OR PROBE H indicates that one of those two operations is being performed.

EN UVECTOR H (not used explicitly by INT chip) indicates that the hardware is reading the UVECTOR lines, between macro-instructions, for PTE CHECK or PROBE, UTRAP, REI check, or interrupt.

BUT UVECTOR H (not used by INT chip) indicates that the UVECTOR lines are being read during an interruptable macro-instruction (for interrupt service).

UVECTOR BRANCH H is the OR of the above two signals. It indicates that the micro-vector lines are being read, and together with some other signals (see section 2.7.7) indicates that the highest priority pending interrupt is being processed and that we may clear the corresponding internal interrupt latch, if there is one.

Interrupt Signals: Various other request/service/status signals relate to interrupt processing.

SPFI L (Synchronized Power Fail Intpt) is a request for an interrupt at IPL 1 E. This request appears and ends at the trailing edge of MCLK; it is latched internally to produce SPFIR (SPFI Request).

WEI L (Write bus Error Intpt) is a request for an interrupt at IPL 1D. It appears at the trailing edge of BCLK and ends at the trailing edge of the next MCLK; it is latched internally to form WEIR (WEI Request).

CPI L (Cache Parity Intpt) is a request for an interrupt at IPL 1B (to log the cache parity error). It appears at the trailing edge of BCLK and ends at the trailing edge of the next MCLK; it is latched internally to form CPIR (CPI Request). [Note: VAX-11/750 will have this line tied high (disabled).]

CDI L (Corrected Data Intpt) is a request for an interrupt at IPL 1A. It appears at the trailing edge of BCLK and ends at the trailing edge of the next MCLK; it is latched internally to form CDIR (CDI Request).

TIMER INT L is a request from the interval timer for an interrupt at IPL 18. It remains asserted until the timer's status registers are read during the servicing of that interrupt and thus needs not be latched in the INT chip.

SLINE INT L is a request from one of the two serial data lines for an interrupt at IPL 14. It too remains asserted until it is serviced and thus needs not be latched.

SBR7 H thru SBR4 H (Synchronized Bus Request n) are requests, from devices on the Unibus, for use of the Unibus and for processor interrupts at IPL 17 thru 14, respectively. They appear and end at the trailing edge of MCLK.

PROC INIT L (Processor Initialize) is an input signal (generated externally upon decoding the relevant Bus Function code) indicating that all internal registers should be initialized (cleared).

UVECTOR BRANCH H is used in clearing internal interrupt latches, as described above and later in section 2.7.7.

INT PNDG L (Interrupt Pending) is an output indicating to the interrupt handling hardware/microcode that there is an interrupt (at a level higher than the current IPL) waiting to be processed.

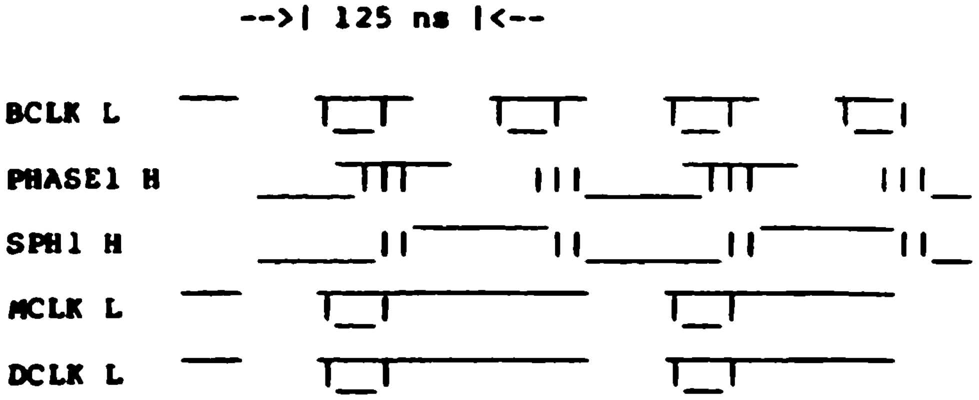
BUS GRANT H is an output to the external Unibus arbitration circuitry instructing it to issue a grant signal (BGn) at the highest pending request level (BRn), as soon as the Unibus becomes free.

HPBG6,5,4 H (Highest Priority Bus Grant n), together with the input signal SBR7 H, identify the highest pending request level. Only one of the four signals (corresponding to the highest request level) will be asserted.

SYNCHR RESET BG H is an input signal from the external Unibus arbitration logic instructing the INT chip to clear the BUS GRANT signal. This reset signal appears synchronously with the first BCLK trailing edge after the start of WAIT (WAIT = SACK + INTR from the Unibus) or the end of the 5-to-10 usec. SACK timeout, which was started at the issuing of the last BGn.

2.3 **Internal Clock Signals:** Input clock signal PHASE1 H is high during the first half of an MCLK cycle. External skews could cause an edge of PHASE 1 H to be seen at INT before the end of BCLK. SPH1 H (Synchronized PHASE1 H) is a signal, generated internally from PHASE1 H and BCLK, which will change state no sooner than two gate delays after the trailing edge of BCLK L.

These timing relationships are shown in INT figure 2.



INT Figure 2 - Timing Diagram (Clock Diagram)

2.4 WBUS/WCTRL Operations (Read/Write PSL):

2.4.1 Relevant Signals:

WCTRL	6 lines	inputs
WBUS	10 lines	i/o 3-st

2.4.2 Registers on Chip: corresp. WBUS (PSL) bits

IS	1 bit	\		26
CURMODE	2 bits	|	part of	25:24
PRVMODE	2 bits	|	PSL reg	23:22
IPL	5 bits	/		20:16

ASTLVL (AST LeVeL)	3 bits	26:24
HSIPR (Highest Software IPR)	5 bits	20:16
LUBIPR (Last granted UniBus IPR)	5 bits	20:16

(All registers consist of transparent latches, except for CURMODE, which uses edge-triggered FFs).

2.4.3 **WBUS Operations:**

I/O	WCTRL code (hex)	generic function	remarks
I	00	PSL <- WBUS	(using 10 bits of WBUS)
O	04	PSL -> WBUS	
I	31	PRVMODE <- WBUS <23:22>	
I	35	PRVMODE <- CURMODE then IS/CURMODE <- WBUS <26:24>	
I	3D	IPL <- WBUS <20:16>	
I	38	ASTLVL <- WBUS <26:24>	
O	3A	ASTLVL -> WBUS <26:24>	
I	3C	HSIPR <- WBUS <20:16>	(see also section 2.7)
O	3F	LUBIPR -> WBUS <20:16>	(see also section 2.8)
O	33	" "	and issue Bus Grant

2.4.4 **Other WCTRL Operations:**

33	Bus Grant	(see section 2.8)
37	REI check	(see section 2.6)
10	IS/CURMODE -> UVECTOR lines	(see section 2.5)

2.4.5 **Operation/Timing:** The WCTRL code is read into transparent latches during SPH1 H assertion (the first half of an MCLK/DCLK cycle), and latched during SPH1 L assertion (the second half). Data steering signals are d.c. decoded from this latched WCTRL code. For operations requiring output onto the WBUS, the appropriate WBUS transmitters (tri-state) are enabled during SPH1 L. For operations requiring input from the WBUS, the relevant WBUS received signals are read into transparent latches (except CURMODE -- see below) on the next DCLK H assertion, which will occur towards the end of SPH1 L. (Similarly, when WCTRL = 35, PRVMODE is loaded from CURMODE on a DCLK; CURMODE edge-triggered FFs do not change until the trailing edge of DCLK.)

2.5 **UVECTOR Operations (outputs):** INT places one of 4 data items onto the UVECTOR lines:

A. Nothing (high impedance) - controlled by status/service signals.

B. Results of REI check (0 R2 R1) - controlled by WCTRL.

C. IS/CURMODE (IS CUR1 CUR0) - controlled by WCTRL.

D. (latched) UVECTOR Interrupt Code - default.

2.5.1 **Relevant Signals:**

WCTRL	6 lines	inputs
UVECTOR	3 lines	outputs, 3-state
DOSRV L		from DPM17
UTRAP L		from MICØ7
PTE CHK OR PROBE H		from MICØ7

2.5.2 **Registers on Chip:**

IS	1 bit \	
CURMODE	2 bits /	part of PSL

Note that INT drives only 3 of the 4 COMET micro-vector lines (because of pin limitations). When INT asserts UVECTOR <2:Ø> H, UVECTOR 3 H will be driven by other COMET circuitry, so that UVECTOR <3:Ø> will not be ambiguous.

2.5.3 **Operations:** if PTE CHK or PROBE => disable UVECTOR outputs (high impedance)
or if (UTRAP & notDOSRV) => disable UVECTOR outputs
or if WCTRL = 37 (REI check) => Ø R1 R2 -> UVECTOR lines
(see section 2.6)
or if WCTRL = 1Ø => IS/CURMODE -> UVECTOR lines
otherwise
if INT PNDG =>
UVIC -> UVECTOR lines (see section 2.7.4)
if not(INT PNDG) =>
fake UVIC (Ø1Ø) -> UVECTOR lines

(performed in priority shown) (see also section 2.8.6)
NOTE: DOSRV always causes UTRAP,

When taking an interrupt between instructions, the processor issues DOSRV & UVECTOR BRANCH H at the time it reads the UVECTOR lines.
When taking an interrupt within an (interruptable) instruction, the processor does not issue DOSRV -- see section 2.7.7.

2.5.4 **Timing:** The input control signals (WCTRL and the 3 service/status signals) change at the trailing edge of MCLK. Any necessary calculations (e.g. REI check) and data switching are then performed. The microcode uses the UVECTOR lines by branching (BUTing) on them at the next MCLK.

Because of timing constraints external to INT, any micro-code branch on UVECTOR lines, will occur in an "extended cycle" (MCLK is stalled T ns, with PHASE1 deasserted, until the time of the ensuing BCLK).

2.6 REI Check: INT performs part of the calculation specified for the VAX REI (Return from Exception or Interrupt) instruction (cf. VAX SRM section 6.13 or VAX 11/780 Architecture Handbook Volume 1 section 12.9) in response to the VAX-11/750 micro-code REI check command.

2.6.1 Relevant Signals:

WCTRL	6 lines	inputs
UVECTOR	4 lines	outputs, 3-st
WBUS	10 lines	i/o, 3-st

2.6.2 Registers on Chip:

IS	1 bit \	
CURMODE	2 bits }	part of PSL
IPL		5 bits /
ASTLVL		3 bits

2.6.3 Operation:

A. continuously calculate R1 = [saved PSL<IS>.EQ.0] &
[saved PSL<CURMODE>.GE.ASTLVL]
(NOTE: CURMODE has 2 bits, ASTLVL has 3; thus ASTLVL MSB (AST 2) must be 0 to set R1 to 1.)

B. continuously calculate
R21 = [saved PSL<CURMODE>.LT.PSL<CURMODE>]
R22 = [saved PSL<IS>.EQ.1] & [PSL<IS>.EQ.0]
R23 = [saved PSL<IS>.EQ.1] & [saved PSL<CURMODE>.NE.0]

R24 = [saved PSL<IS>.EQ.1] & [saved PSL<IPL>.EQ.0]
R25 = [saved PSL<IPL>.GT.0] & [saved PSL<CURMODE>.NE.0]
R26 = [saved PSL<PRVMODE>.LT.[saved PSL<CURMODE>]

R27 = [saved PSL<IPL>.GT.PSL<IPL>]
R2 = R21 + R22 + R23 + R24 + R25 + R26 + R27

where PSL<X> is the X register of section 3.3.6.2 and saved PSL<Y> is on the WBUS (in the position corresponding to its position in the PSL -- cf. section 3.3.4.2)

C. if WCTRL = 37 (hex) => do the following:
a) remove UVIC from UVECTOR lines
b) put R1 and R2 on lines UVECTOR 0 H and UVECTOR 1 H, respectively; put 0 on line UVECTOR 2 H

2.6.4 **Timing:** The values R1, R21 thru R27, and R2 are continuously calculated from the input signals from the WBUS receiver gates and the PSL and ASTLVL latches. After MCLK, WCTRL latches stabilize (during SPH1 H) and 0 R2 R1 are placed onto the UVECTOR lines. After the next MCLK reads the next micrq-code instruction, the (presumably) different WCTRL code causes 0 R2 R1 to be removed.

2.7 Interrupt Arbitration (INT PNDG gen'n) and Granting:

2.7.1 Relevant Signals:

IPL	I/O	name			
1E	I	SPFI L	synchr power fail intpt		$
1D	I	WEI L	write error intpt		$
1B	I	CPI L	cache parity intpt		$
1A	I	CDI L	corrected data intpt		$
18	I	TIMER INT L	interval timer intpt		@
14	I	SLINE INT L	serial line intpt	*	@
14-17	I	SBRn H	synchr bus req.	&	@
	O	INT PNDG L			

* -- there are two serial lines (TTY, cassette); their interrupts will be ORed together externally to INT

& -- a multiprocessor, if present, will hang on one of the BR lines

$ -- see para. 2.7.3 -- must latch

@ -- see para. 2.7.3 -- static

2.7.2 Registers on Chip:

IPL	name		
1E	SPFIR	(1 bit)	latch for SPFI
1D	WEIR	(1 bit)	latch for WEI
1B	CPIR	(1 bit)	latch for CPI
1A	CDIR	(1 bit)	latch for CDI
01-0F	HSIPR	highest software IPR	(5 bits)
00-1F	IPL		

The first four registers are SR latches. The last two are (clocked) transparent latches.

2.7.3 **Latched and Static Interrupts:** Four inputs (SPFI, WEI, CPI, CDI) begin at a BCLK (MCLK, in the case of SPFI) trailing edge and end at the next MCLK trailing edge. They are thus present for only T or 2T nsec (thru one MCLK) and must therefore be latched into 1-bit SR latches (SPFIR, WEIR, CPIR, CDIR).

Two inputs (TIMER INT, SLINE INT) are static and need not be latched. SLINE INT begins at an MCLK trailing edge; TIMER INT begins at a DCLK trailing edge.

The four SBRn come from external synchronizing FFs which are clocked on the trailing edge of MCLK.

2.7.4 **UVIC Assignments:** The UVIC (UVECTOR Interrupt Code) identifies the highest priority interrupt request present. It is d.c. encoded from the interrupt inputs and is latched (into edge-triggered FFS) on an MCLK trailing eoge, to store the LATCHED UVIC, which is (sometimes) placed onto the UVECTOR lines. (Similarly, interrupt requests are continuously compared vs. current IPL to produce the internal signal INT PNDG H, which is latched into an edge-triggered FF an each MCLK trailing edge. LATCHEC INT PNDG thus corresponds to LATCHED UVIC. The INT output signal INT PNDG L is actually the LATCHED version, so that it corresponds to LATCHED UVIC on the UVECTOR lines.)

The UVIC values are assigned to interrupt requesters in order of IPL, as indicated herein.

IPL	name		UVIC
1E	SPFIR	synchr power fail intpt req	111
1D	WEIR		110
1B	CPIR		101
1A	CDIR		100
18	TIMER INT	interval timer intpt	011
14-17	SBRn	bus req. (any one asserted)	010
14	SLINE INT	serial line intpt	001
01-0F	HSIPR (non-zero)		000
00	no interrupt request present		000

2.7.5 **The Interrupt Procedure:** One or more of the 10 hardware interrupt lines may be asserted, or there may be a software interrupt request pending. The microcode keeps track of software interrupt requests and keeps the INT chip informed of the IPL of the highest one pending by writing into the HSIPR.

Within INT, the identity of the highest level interrupt request is encoded as the (latched) UVIC, which is updated every MCLK. The UVIC value is established independently of the current IPL (so that the interrupter indicated by the UVIC is not necessarily at a higher IPL than the current program). If there is an interrupt request at a level higher than the present IPL, INT PNDG L will be asserted. (Similarly to LATCHED UVIC, INT PNDG L is updated at each MCLK trailing edge.) Other COMET hardware checks INT PNDG between macro-instructions, and the microcode checks it during interruptable macro instructions. In either case, if INT PNDG is asserted, the microcode will branch (BUT) on the UVECTOR lines, which will be indicating the LATCHED UVIC, to a micro-code subroutine that will process the indicated interrupt.

All interrupters are uniquely identified by the UVIC with the exceptions of Unibus requests and software interrupt requests. In the case of the latter, the microcode already knows the IPL of the highest pending software interrupt, so that if UVIC = 000, it knows what to do. If (after seeing INT PNDG) the microcode sees UVIC = 010, it will order (via WCTRL = 33) the issuing of a Bus Grant and will read the IPL of the grantee from the LUBIPR (via the WBUS), as described in sections 2.4.3 and 2.8.4.

2.7.6 Arbitration:

UVIC Calculation: The eleven interrupt lines/internal registers (all of which change state on an MCLK (or DCLK) trailing edge) are continuously monitored to compute the interrupt code, which identifies the highest pending interrupt request, regardless of whether or not its level is higher than the current IPL. The computed code is then loaded, at each MCLK trailing edge, into a register of edge triggered FFs, whose output is the LATCHED UVIC.

If none of the higher priority UVECTOR operations is in effect (see section 2.5.3), and if INT PNDG is asserted, the LATCHED UVIC is placed onto the UVECTOR lines. But if INT PNDG is not asserted, the default value 010 (Unibus Interrupt request) is placed onto the UVECTOR lines, for reasons discussed in section 8 under "fake grants".

INT PNDG Calculation: The output signal INT PNDG signifies that an interrupt request is present at a level higher than the current IPL. For interrupters with codes other than 000 and 010, a combinatorial function of (unlatched) interrupt code and IPL easily determines whether INT PNDG should be asserted. For codes 000 and 010 the procedure is more complicated.

The value of UBIPR<1:0> (calculated as per section 2.8.3) identifies the IPL of the highest pending BR. The default value (with no BR present) is UBIPR<1:0> = 00, which looks like the BR4 encoding. However, the signal UBI (= SBR7 + SBR6 + SBR5 + SBR4) indicates the presence of a real BR, so that a logic function of UBIPR<1:0>, IPL, and UBI determines whether INT PNDG should be asserted for a Unibus requester. (Note that this function is independent of UVECTOR interrupt code.)

A 5-bit magnitude comparator is used to compute the logic function (HSIPR .GT. IPL), which determines whether INT PNDG should be asserted for a software interrupt requester. (Again this function is independent of UVECTOR interrupt code.)

These calculations are done using the unlatched interrupt code, so that the next MCLK trailing edge will latch UVIC and the corresponding INT PNDG value at the same time.

2.7.7 Clearing Interrupt Requests: Each of the four latched interrupt registers (SPFIR, WEIR, CPIR, CDIR) must be cleared when the corresponding interrupt is taken. The reading of the UVECTOR lines is signified by the assertion of input signal UVECTOR BRANCH H. This reading might be for one of the following reasons : UTRAP, PTE CHECK OR PROBE, REI CHECK (signified by WCTRL = 37), IS/CUR (signified by WCTRL = 10), or taking an interrupt. Thus the internally generated signal UV INTPT = UVECTOR BRANCH & notUTRAP & not(PTE CHECK OR PROBE) & not(SEND REI) & not(SEND IS/CUR) signifies the taking of an interrupt.

Because the UTRAP signal is valid only at about the time of MCLK, we must do the condition checking and latch clearing at that time (CLR INT LATCHES H = MCLK H & UV INTPT H). The CLR INT LATCHES H pulse is steered to the correct latch by the current value of (latched) UVIC, which identifies the interrupt being taken.

Since these latches are cleared during MCLK, and their respective setting inputs end at MCLK (cf section 3.3.7.3), we could miss flagging a new interrupt while clearing the latch for a previous one. This is not a serious problem, however because:

A) CDI is just for error logging. If such interrupts come so frequently that we miss every other one, the logging routine will still get the message.

B) CPI -- ditto. (In fact, for COMET, cache parity error has been changed from an interrupt to an exception, so that this latch will never be set.

C) WEI is a serious error. Once the first such interrupt is taken, IPL is raised to 1D and further "normal" CPU operation is aborted anyway. A second WEI is thus redundant.

(As a matter of interest, system delay times are such that an ensuing WEI signal will probably remain asserted after the clear pulse has ended anyway.)

D) SPFI is again serious. The power failure routine will raise IPL to 1E to ignore further SPFIs anyway.

Each of the interrupts TIMER INT and SLINE INT is cleared by the requesting device itself when the corresponding interrupt is taken (when the device detects that its status register has been read).

Each Unibus device clears its own BRn when it receives the BGn signal granting it the Unibus interrupt (unless it wishes to continue interrupting, in which case it continues to assert BRn).

Finally, when the microcode takes a software interrupt it automatically updates the HSIPR to reflect that fact.

2.8 Unibus Request/Grant:

2.8.1 Relevant Signals:

I/O	name	descr.	
I	SBRn	synchr bus req.	4 lines
I	WCTRL		6 lines
O	HPBGn	highest priority bus grant	3 lines
O	BUS GRANT H		
I	SYNCHR RESET BG H	reset Bus Grant	
O	WBUS		10 lines

2.8.2 Registers on Chip:

IPL	name		
14-17	LUBIPR	Last granted UniBus IPR	(5 bits)

2.8.3 **Requesting:** As indicated in section 2.7.6, if the highest pending interrupt request is from a Unibus device (or if there is <u>no</u> interrupt pending that is higher than the current IPL) the UVIC value for Unibus Requests (010) is placed onto the UVECTOR lines. If the microcode sees INT PNDG L asserted and then reads

UVECTOR = 010, it assumes a Unibus device deserves an interrupt and will issue WCTRL = 33 to cause a Bus Grant (see section 2.8.4).

The incoming Synchronized Bus Request lines, SBR7 thru 4, are used to generate the two-bit value UBIPR<1:0> H, which is the binary value of the number of the highest pending Bus Request (note that UBIPR is independent of processor IPL). The default value (no pending SBRs) is UBIPR<1:0> = 00. [i.e. UBIPR1 = SBR7 + SBR6; UBIPR0 = SBR7 + (SBR5 & notSBR6)]

A combinatorial function of UBIPR and the current IPL produces the signal (BR > IPL). Furthermore, the signal UBI = SBR7 + SBR6 + SBR5 + SBR4 tells us if there really is a Bus Request present. Finally, EN HPBG H = (BR > IPL) & UBI indicates whether there is a pending Bus Request that deserves service because it is at a level higher than the IPL. EN HPBG H is one of the signals that causes generation of INT PNDG.

2.8.4 Granting: After seeing INT PNDG asserted, then UVIC = 010, the microcode will order the issuing of a Unibus Grant (at the level determined by the INT chip) by setting WCTRL = 33. As described in section 2.4.5, the received WCTRL code stabilizes during SPH1 H (the first half of a microcycle). If WCTRL = 33, the internal signal ISSUE BG is asserted. (ISSUE BG & SPH1) is then the D input to the edge-triggered BG FF that loads on the leading edge of BCLK, so that the FF will be set in the middle of a microcycle where WCTRL = 33 has been received. The FF output is ANDed with SPH1 L to produce chip output signal BUS GRANT H, a signal that starts just after the BCLK in the middle of a microcycle (see section 2.3 for a description of these timing signals). Furthermore, the clock stalling circuitry (another part of the CPU), after seeing WCTRL = 33, will stall (disable) MCLK and PHASE1 (both deasserted).

The assertion of BG enables the inputs to two transparent latches that then store LUBIPR<1:0> (Last granted UBIPR) until the next Bus Grant is issued. This LUBIPR is then put onto the UVECTOR lines during this Bus Grant (WCTRL = 33) micro-cycle. (WCTRL = 37 would order the placing of LUBIPR onto the UVECTOR lines without updating it, nor issuing Bus Grant).

Externally to the INT chip, BUS GRANT H performs two functions. It is arbitrated (in the Unibus arbitration circuitry) against NPR, NPG, and SACK to assert BG EN, the enable signal to the four Unibus BG drivers, at the appropriate time; and its trailing edge causes the release of the stalled MCLK and PHASE1.

Internal signals HPBR6 thru 4, along with SBR7, denote the Highest Priority pending Bus Request (independent of current IPL). Thus at most one of these four signals will be asserted at any given time. HPBR6 thru 4 are then ANDed with EN BG H (see section 2.8.3) to form output signals HPBG6 thru 4 H, which then go to the Unibus BG drivers, along with SBR7 H. Thus one of the three Unibus Grant lines BG6 thru 4 may be asserted only if we know, from EN BG H, that the request at that level is indeed higher than the current IPL. We may issue a BG7 without explicitly testing EN BG H, because we know that, if an SBR7 is present at the time of the issuing (by microcode) of BUS GRANT, it is indeed at a level higher than the present IPL (see section 2.8.6).

2.8.5 Clearing: The input signal SYNCHR RESET BG H is used to clear the BG FF (at the next BCLK leading edge), thereby ending BUS GRANT and releasing the clock stall. SYNCHR RESET BG H is a synchronized signal, starting at an MCLK trailing edge, to avoid oscillation of the BG FF that would be caused by changing this reset signal at the same time as the FF's clock edge. Conditions causing the assertion of SYNCHR RESET BG H are the assertion of WAIT (effectively equal to SACK, for this purpose) on the Unibus, or the assertion of NO SACK TIMEOUT in the Unibus arbitration circuitry.

2.8.6 Fake Grants: It is possible that an interrupt request will be asserted then removed before being serviced. (Only a Unibus device will ever commit this offense.) If this request is the highest one pending, is at a level higher than the current IPL, and is present thru MCLK, it will cause assertion of INT PNDG L, and UVIC will change (to 010) to indicate its identity. If now the request is removed after INT PNDG L has been read (by micro-code or hardware), but before UVIC has been read, UVIC will revert to its previous value and may now identify a requester at a level below the current IPL. In such a case, the deassertion of INT PNDG forces the value 010 (fake BR) onto the UVECTOR lines. Thus anytime INT knows there is no pending interrupt (> IPL) but the micro-code thinks there is and branches on UVECTOR lines to service it, it (micro-code) will be told that the interrupt request is a BR. (The micro-code does not know or care which BR, and in fact IPL must be less than 17, so that some level BG is permissible, because only a Unibus (BR) device could have asserted and removed its request.) Now micro-code will issue WCTRL = 33, ordering a Bus Grant. INT knows that no present BR is above IPL and therefore forces HPBG6,5,4 H to 0 to prevent issuing Unibus BG6,5, or 4. Furthermore, SBR7 is not asserted (if it were, we would have a legitimate interrupt and wouldn't be in this mess, because we know IPL < 17), so that Unious BG7 will not be asserted. However, we must issue BUS GRANT (which will ultimately enable the Unibus BG drivers to transmit their 0s) to release the stalled MCLK. No BGn results in no SACK, thus SYNCHR RESET BG H would normally end BUS GRANT (at the first BCLK leading edge after the first MCLK trailing edge) after the assertion of the NO SACK TIMEOUT. However, INT knows this is a fake grant and can therefore end BUS GRANT early.

The deassertion of EN HPBG (see section 2.8.3) AND the deassertion of SPH1 will cause the BG FF to clear at the next BCLK leading edge. (The BG FF was allowed to set, despite the deassertion of EN HPBG, on the leading edge of BCLK during the assertion of SPH1.) Thus Bus Grant will last just long enough (T ns) to release the clock stall, then operations will resume.

```
                     ---------------
           WEI L --->!01         48!<--- TIMER INT L
           CPI L --->!02         47!---> INT PNDG L
          SPFI L --->!03         46!---> BUS GRANT H
           CDI L --->!04         45!<--- SBR4 H
       WCTRL 4 H --->!05         44!---> HPBG6 H
       WCTRL 5 H --->!06         43!---> HPBG4 H
        PHASE1 H --->!07         42!<--- SBR5 H
       WBUS 22 H <-->!08         41!---> HPBG5 H
       WBUS 23 H <-->!09         40!<--- SBR7 H
     PROC INIT L --->!10 ....... 39!<--- SBR6 H
       WCTRL 2 H --->!11 .     . 38!---- GROUND
             VGA ----!12 . LID . 37!<--- SYNCHR RESET BG H
             VCC ----!13 . DOWN. 36!<--- BCLK L
       WCTRL 1 H --->!14 .     . 35!---- GROUND
       WCTRL 3 H --->!15 ....... 34!<--- MCLK EN H
       WBUS 25 H <-->!16         33!<--> WBUS 16 H
       WBUS 24 H <-->!17         32!<--> WBUS 17 H
       WBUS 26 H <-->!18         31!<--> WBUS 19 H
       WCTRL 0 H --->!19         30!<--- SLINE INT L
     UVECTOR 0 H <---!20         29!<--- DCLK EN H
     UVECTOR 2 H <---!21         28!<--> WBUS 18 H
     UVECTOR 1 H <---!22         27!<--> WBUS 20 H
        UTRAP L  --->!23         26!<--- PTE CHK OR PROBE H
UVECTOR BRANCH H --->!24         25!<--- DOSRV L
                     ---------------
```

INT FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.12 INSTRUCTION REGISTER DECODE (IRD DC622)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for the VAX-11/750 Instruction Register Decode.

IRD Table 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE	PIN #	PIN ID	COMMENTS	GATE TYPE
01	XB 00 H		GA1TNF	25	CS ADDR 02 L	I_{OL}=12MA	GA1TCN
02	XB 04 H		GA1TNF	26	CS ADDR 01 L	I_{OL}=12MA	GA1TCN
03	XB 01 H		GA1TNF	27	REG MODE H		GA1TTN
04	XB 10 H		GA1TZF	28	CS ADDR 00 L	I_{OL}=12MA	GA1TCN
05	IR 1 H		GA1TTN	29	CS ADDR 03 L	I_{OL}=12MA	GA1TCN
06	IR 6 H		GA1TTN	30	IRD CONTROL L		GA1TNF
07	IR 7 H		GA1TTN	31	IRD ADD CTL 1 H		GA1TNF
08	XB 02 H		GA1TNF	32	M CLK L		GA1TNF
09	IR 5 H		GA1TTN	33	XB 14 H		GA1TZF
10	IR 2 H		GA1TTN	34	IRD ADD CTL 0 H		GA1TNF
11	XB 09 H		GA1TZF	36	IRD RNUM 3 H		GA1TTN
14	IR 4 H		GA1TTN	37	IRD RNUM 0 H		GA1TTN
15	DST RMODE H		GA1TTN	39	IR 3 H		GA1TTN
16	PSL CM H		GA1TNF	40	IRD RNUM 1 H		GA1TTN
17	XB 05 H		GA1TNF	41	IRD RNUM 2 H		GA1TTN
18	LD OSR L		GA1TNF	42	XB 06 H		GA1TNF
19	XB 15 H		GA1TZF	43	XB 03 H		GA1TNF
20	XB 12 H		GA1TZF	44	DISP ISIZE 1 H		GA1TTN
21	DISP ISIZE 0 H		GA1TTN	45	WCTRL 2 H		GA1TNF
22	XB 13 H		GA1TZF	46	XB 08 H		GA1TZF
23	LD IR L		GA1TNF	47	XB 11 H		GAJTZF
24	XB 07 H		GA1TNF	48	IR 0 H		GA1TTN

2. **Performance:**

2.1 IRD State Elements:

2.1.1 INSTR REG <7:0>: This is an 8 bit register built from latches. Clocking of the latches is enabled when LD IR L is asserted. If clocking is enabled, the latches open when M CLK L is asserted. When M CLK L is not asserted the latches are closed. The INSTR REG is loaded with XB <07:00> H if PSL CM H is not asserted, and with XB<15:08> if PSL CM H is asserted. There is no way to directly clear this register.

2.1.2 OSR <7:0>: This is an 8 bit register built from D type flip flops. It is clocked by the rising edge of M CLK L whenever LD OSR L is asserted. The data is specified herein:

PSL CM H	LD IR L	Data Source
L	H	XB <07:00> H
L	L	XB <15:08> H
H	X	XB <07:00> H

2.2 Pin Descriptions:

2.2.1 M CLK L: This is the clock associated with the micro-sequencer. M CLK L is used to latch the INSTR REG and to clock the OSR.

2.2.2 LD IR L: This signal specifies when an opcode is being fetched. When LD IR L is asserted the INSTR Reg is loaded with XB <07:00> or XB <15:08>. It is also used to generate IR <7:0> H, CS ADDR <03:00> L, IRD RNUM <3:0> H, and REG MODE H.

2.2.3 LD OSR L: This signal specifies that an operand specifier is being evaluated. When LD OSR L is asserted the OSR is loaded with XB <15:08> or XB <07:00>. It is also used to generate IR <7:0> H, CS ADDR <03:00> L, IRD RNUM <3:0> H, and REG MODE H.

2.2.4 PSL CM H: This signal indicates which mode of instructions the machine is executing, Native or Compatibility mode. The state of this signal determines what is loaded into the INSTR REG and the OSR. It is also used to generate all output signals except XB <15:08> H.

2.2.5 IRD ADD CTL <1:0> H: These signals specify what branch information should be ORed onto CS ADDR <03:00> L. See description of CS ADDR <03:00> L, Paragraph 2.2.11.

2.2.6 **IRD CONTROL L:** This signal enables the INSTR REG or the OSR onto XB <15:08>. (See Paragraph 2.2.9).

2.2.7 **WCTRL 2 H:** This signal selects whether the INSTR REG or the OSR should be read onto XB <15:08>. See Paragraph 2.2.9.

2.2.8 **XB <07:00> H:** This half of the XB lines are used as input only. They contain an opcode, an operand specifier, or part of a Compatibility mode instruction. XB <07:00> H are used to generate.

IR <7:0> H
CS ADDR <03:00> L
IRD RNUM <3:0> H
REG MODE H

2.2.9 **XB <15:08>:** The upper half of the XB lines are used as both input and output. When used as input, the signals contain an operand specifier or the opcode portion of a Compatibility mode instruction. Besides being used as data for the INSTR REG and the OSR it is used to generate

IR <7:0> H
CS ADDR <03:00> L
IRD RNUM <3:0> H
REG MODE H

When used as output, XB <15:08> H are driven as follows:

IRD CONTROL L	WCTRL 2 H	XB <15:08> H
H	X	Z (High Inpedance)
L	L	INSTR REG <7:0> H
L		H OSR <7:0> H

2.2.10 **IR <7:0> H:** This group of signals contains the opcode or an encoding of the opcode which is used by the micro-sequencer. They are specified herein. See IRD Table 2.

PSL CM H	LD IR L	IR <7:0> H
L	L	XB <07:00> H
L	H	INSTR REG <7:0> H
H	L	See Table III where Opcode <15:00> = XB <15:00> H
H	H	See Table III where Opcode <15:08> = INST REG <7:0> H

Opcode <Ø7:ØØ> = OSR
<7:Ø> H

IRD Table 2 IR <7:Ø> H

Decode Class	Instr. Class	IR 7	IR 6	IR 5	IR 4	IR 3	IR 2	IR 1	IR O
Opcode<14:11>=ØØØ1	A	H	L	Opcode<8>	Opcode<7>	Opcode<15>	Opcode<1Ø>	Opcode<Ø9>	Opcode<Ø6>
Opcode<14:12>≠Ø and Opcode<14:12>≠7	B1	L	H	L	Opcode<7>	Opcode<15>	Opcode<14>	Opcode<13>	Opcode<12>
Opcode<14:11>=111Ø or 1111	B2	L	H	H	Opcode<7>	Opcode<15>	Opcode<1Ø>	Opcode<Ø9>	Opcode<11>
Opcode<14:11>=Ø and Opcode<15><1Ø:Ø8>≠Ø	C	L	L	L	H	Opcode<15>	Opcode<1Ø>	Opcode<Ø9>	Opcode<Ø8>
Opcode<15:Ø8>=Ø and Opcode<Ø7:Ø6>=Ø	D1	H	H	L	Opcode<7>	Opcode<15>	Opcode<Ø2>	Opcode<Ø1>	Opcode<ØØ>
Opcode<15:Ø8>=Ø and Opcode<Ø7:Ø6>=1,2,or 3	D2	H	H	H	Opcode<7>	Opcode<15>	Opcode<Ø4>	Opcode<Ø5>	Opcode<Ø6>

2.2.11 **CS ADDR <03:00> L:** These are the lowest four Control Store Address lines. They are driven based on a decode of the operand specifier being evaluated. They can also be driven as a function of INSTR REG <2:0>. See IRD chart 1.

2.2.12 **IRD RNUM <3:0> H:** These signals specify the register number associated with the operand specifier. They are generated as shown in IRD chart 2.

2.2.13 **REG MODE H:** This signal indicates whether the operand specifier being evaluated specifies register mode or not. See IRD chart 3.

IRD CHART 1

IRD ADD CTL <1:0> H	PSL CM H	LD IR L	INSTRUCTION CLASS	CS ADDR <03:00> L
0	X	X	X	1111 (No branch)
1	L	L	X	See chart 1A where AMODE <3:0> = XB <15:12> H REG NUM <3:0> = XB <11:08> H
1	L	H	X	See chart 1A where AMODE <3:0> = XB <07:04> H REG NUM <3:0> = XB <03:00> H
1	H	L	A,D2, B2(XB <11:09> ≠ 4 or 7)	See chart 1B where AMODE <2:0> = XB <5:3> H REG NUM <2:0> = XB <2:0> H
1	H	L	B1, B2(XB <11:09> = 4 or 7)	See chart 1B where AMODE <2:0> = XB <11:09> H REG NUM <2:0> = XB <08:06> H
1	H	L	C, D1	0001 (Actual Value)
2	L	X	X	See chart 1A where AMODE <3:0> = OSR <7:4> H REG NUM <3:0> = OSR <3:0> H
2	H	X	B1, B2(INSTR REG <3:1> = 4)	See chart 1B where AMODE <2:0> = OSR <5:3> H REG NUM <2:0> = OSR <2:0> H
2	H	X	OTHER	1111 (No branch)
3	X	X	X	1, INSTR REG <2:0> L

IRD CHART 1A

		(Actual Value)
AMODE <3:0>	REG NUM <3:0>	CS ADDR <03:00> L
0,1,2,3	X	1100
4	0 - 14	0010
4	15	0011
5	X	1111
6	X	0111
7	X	1011
8	0 - 14	1110
8	15	1101
9	0 - 14	0100
9	15	1000
10, 2,14	0 - 14	1001
10,12,14	15	1010
11,13,15	0 - 14	0101
11,13,15	15	0110

IRD CHART 1B

AMODE <2:0>	REG NUM <2:0>	(Actual Value) CS ADDR <03:00> L
0	0 - 6	1111
0	7	1110
1	0 - 6	1101
1	7	1100
2	0 - 5	1011
2	6	1010
2	7	0000
3	0 - 6	1001
3	7	1000
4	0 - 5	0111
4	6	0110
4	7	0100
5	X	0101
6	0 - 6	0011
6	7	0010
7	X	0001

IRD CHART 2

PSL CM H	LD IR L	INSTRUCTION CLASS	IRD RNUM <3:0 H
L	L	X	XB <11:08> H
L	H	X	XB <03:00> H
H	L	A, D2, B2(XB <11:09> ≠ 4 or 7)	0, XB <02:00> H
H	L	B1, B2(XB <11:09> = 4 OR 7)	0, XB <08:06> H
H	X	C, D1	0
H	H	A, D2, B2(INSTR REG <3:1> ≠ 4 or 7)	0, INSTR REG <0> H, OSR <7:6> H
H	H	B1, B2(INSTR REG <3:1> = 4 OR 7)	0, OSR <2:0> H

IRD CHART 3

PSL CM H	LD IR L	IRD ADD CTL <1:0> H	INSTRUCTION CLASS	REG MODE H
L	X	X	X	1 if XB <07:04> = 5
H	L	0,1,3	A, D2, B2(XB <11:09> ≠ 4 or 7)	1 if XB <05:03> = 0
H	L	0,1,3	B1, B2(XB <11:09> = 4 or 7)	1 if XB <11:09> = 0
H	X	0,1,3	D1, C	0
H	H	2	X	1 if OSR <5:3> = 0
H	L	2	A, D2 B2(XB <11:09> ≠ 4 or 7)	1 if (XB <05:03> AND OSR <5:3>) = 0
H	L	2	B1 B2 (XB <11:09> = 4 or 7)	1 if (XB <11:09> AND OSR <5:3>) = 0
H	L	2	D1, C	1 if OSR <5:3> = 0
	OTHERWISE			0

2.2.14 **DST RMODE H:** This signal specifies whether the destination operand is register mode or not.

PSL CM H	OSR 7 H	OSR 6 H	OSR 5 H	OSR 4 H	OSR 3 H	DST RMODE H
L	L	H	L	H	X	H
L		OTHERWISE			X	L
H	X	X	L	L	L	H
H	X	X		OTHERWISE		L

2.2.15 **DISP ISIZE <1:0> H:** These signals are used to indicate the size of an address displacement in the ISTREAM. They are specified as follows:

PSL CM H	OSR <7:4> H	DISP ISIZE <1:0> H
H	X	0
L	10,11	1
L	12,13	2
L	14,15	3
L	OTHER	0

```
                             ---------------
          XB 00 H --->|01          48|---> IR 0 H
          XB 04 H --->|02          47|<--> XB 11 H
          XB 01 H --->|03          46|<--> XB 08 H
          XB 10 H <-->|04          45|<--- WCTRL 2 H
           IR 1 H <---|05          44|---> DISP ISIZE 1 H
           IR 6 H <---|06          43|<--- XB 03 H
           IR 7 H <---|07          42|<--- XB 06 H
          XB 02 H --->|08          41|---> IRD RNUM 2 H
           IR 5 H <---|09          40|---> IRD RNUM 1 H
           IR 2 H <---|10 ....... 39|---> IR 3 H
          XB 09 H <-->|11 .     . 38|---- GROUND
              VGA ----|12 . LID . 37|---> IRD RNUM 0 H
              VCC ----|13 . DOWN. 36|---> IRD RNUM 3 H
           IR 4 H <---|14 .     . 35|---- GROUND
      DST RMODE H <---|15 ....... 34|<--- IRD ADD CTL 0 H
         PSL CM H --->|16          33|<--> XB 14 H
          XB 05 H --->|17          32|<--- M CLK L
         LD OSR L --->|18          31|<--- IRD ADD CTL 1 H
          XB 15 H <-->|19          30|<--- IRD CONTROL L
          XB 12 H <-->|20          29|---> CS ADDR 03 L
   DISP ISIZE 0 H <---|21          28|---> CS ADDR 00 L
          XB 13 H <-->|22          27|---> REG MODE H
          LD IR L --->|23          26|---> CS ADDR 01 L
          XB 07 H --->|24          25|---> CS ADDR 02 L
                             ---------------
```

IRD FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.13 MASSBUS CMI INTERFACE CONTROL CHIP (MCI)

1.0 A GENERAL DESCRIPTION

MCI controls the MASSBUS CMI interface. It handles arbitration, command/address control, status generatior & checking, and interrupts. It contains bits of the control, status, and maintenance registers.

2. TERMINOLOGY

Some state elements are defined in terms of signing prefixed by LOAD, CLR, COUNT. If none of these signals are true, the element does not change state.

3. SIGNAL DESCRIPTIONS

3.1 ATTN - ATTN SYNC H

ATTN SYNC H is a TTL input.
ATTN is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
ATTN <---- ATTN SYNC H
```

3.1.1 IE

IE is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
LOAD IE = LD CTRL H
IE <---- IBUS 2 H
```

CLR IE = INIT H
LOAD IE and CLR IE both being true is an illegal control state.

3.1.2 DT BUSY H, INTER L

DT BUSY H and INTER L are TTL inputs

3.1.3 BR, BR L

BR is a transparent latch, open when MBA CLK L is low

```
BR <---- (DT BUSY L   ATTN H + INTER H)* IE H
```

BR L is a totem pole output, and is the low true output of the BR latch.

3.2 BG SYNC L, DO VECTOR CYC

BG SYNC L is a TTL input.

DO VECTOR CYC is an edge triggered D flipflop clocked on the rising edge of MBA CLK L.

DO VECTOR CYC <---- BG SYNC H * BR H * DO CMI CYC L + WAIT H * INIT L

3.2.1 WAIT, WAIT L, WAITING

WAIT is a transparent latch, open when MBA CLK L is low.

WAIT <---- DO VECTOR CYC H * MAST DAT CYC L

WAIT L is an open collector output, and is the low true output of the WAIT latch.

WAITING is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

WAITING <---- DO VECTOR CYC H MAST DAT CYC L

3.2.2 VECTOR CYC, VECTOR CYC L

VECTOR CYC is a set reset latch.

```
SET VECTOR CYC  = WAITING H * CMI CMD EN H
CLR VECTOR CYC  = WAITING L
```

VECTOR CYC L is a totem pole output, and is the low true output of VECTOR CYC.

3.2.3 SLOW CMI L

SLOW CMI L is a TTL input

3.2.4 DO CMI MASTER H

DO CMI MASTER H is an open collector transceiver.

DO CMI MASTER H is driven low by MCI if DRIVE DO CMI MAST H.

DRIVE DO CMI MASTER H = DO VECTOR CYC L * (MAST DAT CYC H + DO CMI CYC L + SLOW CMI H * DLY CNTR EQ 27 H)

3.2.5 ARB, ARB L

ARB is a transparent latch, open when MBA CLK L is low.

ARB <---- DRIVE DO CMI MASTER L

ARB L is a totem pole output, and is the low true output of this ARB latch.

3.3 BLK CMD

BLK CMD is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD BLK CMD = LD MAINT H

BLK CMD <---- IBUS 28H

CLR BLK CMD = INIT H

LOAD BLK CMD and CLR BLK CMD both being true is an illegal condition.

3.4 MAST DATA CYC

MAST DATA CYC is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET MAST DATA CYC = CMI CMD EN H
CLR MAST DATA CYC = DBBZ L

3.4.1 DRIVE DO CMI CYC

DRIVE DO CMI CYC is an edge triggered D flipflop clocked on the rising edge of MBA CLK L.

DRIVE DO CMI CYC <---- BLK CMD H * MM H + DBBZ L * DO VECTOR CYC L * MAST DATA CYC H.

3.4.2 DO CMI CYC H

DO CMI CYC H is an open collector transceiver. DO CMI CYC H is driven low by MCI if DRIVE O CMI CYC H.

3.5 BYTE ROTATE, BYTE ROTATE H

BYTE ROTATE is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD BYTE ROTATE = LD VAR H
BYTE ROTATE <---- IBUS Ø H
BYTE ROTATE H is a totem pole output.

3.6 CMI CMD EN, CMI CMD EN L

CMI CMD EN is an edge triggered D flipflop clocked on the rising edge of MBA CLK L.

CMI CMD EN <---- DO CMI MASTER H * DBBZ L * INIT L

CMI CMD EN L is a totem pole output

3.6.1 CMI OUT EN

CMI OUT EN is a transparent D latch, open when MBA CLK L is low.

CMI OUT EN <---- CMI CMD EN H

3.6.2 CMI <27:25> H

CMI <27:25> H are three tri-state transceivers. They are enabled if CMI OUT EN H.

CMI 27 H <---- (DT FUN 1 L * DT FUN 0 H) + DO VECTOR CYC L

CMI 26 H <---- DO VECTOR CYC L

CMI 25 H <----- 0

3.6.3 CMI FUN <2:0>

CMI FUN <2:0> are three transparent latches, open if ADD CYC POS H * MBA CLK H

CMI FUN <2:0> <---- CMI <27:25> H, respectively.

3.7 DRIVE DBBZ, DBBZ L

DRIVE DBBZ is a transparent D latch, open when MBA CLK L is low.

DRIVE DBBZ <---- CMI CMD EN H + ADD ERROR L * DO SLAVE OP H
* SLAVE DONE L.

DBBZ L is an open collector transceiver. MCI drives DBBZ L low if DRIVE DBBZ H.

3.8 SLAVE DONE H, ADD ERROR L

SLAVE DONE H and ADD ERROR L are TTL inputs.

3.8.1 CHECK FUN

CHECK FUN <---- DBBZ H + ADD OK H * ADD ERROR L * SLAVE DONE L.

3.8.2 ADD OK H

ADD OK H is an open collector transceiver,
ADD OK H is driven low by MCI if
(CMI FUN 1 H * (CMI FUN <2:0> = 2) + CHECK FUN L

3.8.3 DO SLAVE OP

DO SLAVE OP is an edge triggered D flip flop, clocked on the rising edge of MBA CLK L.
DO SLAVE OP <---- ADD OK H.

3.8.4 SLAVE READ L

SLAVE READ L is a totem pole output.

SLAVE READ L = DO SLAVE OP L + ADD ERROR H + CMI FUN 2 H

3.9 ADD CYC POS

ADD CYC POS is an edge triggered D flipflop clocked on the rising edge of MBA CLK L.

ADD CYC POS <---- DBBZ L

3.10 STAT <1:0>

STAT <1:0> are two transparent D latches, open when MBA CLK L is low.

STAT 1 <---- ADD ERROR L * RETURN STAT L

STAT 0 <---- RETURN STAT L

3.10.1 STAT OUT EN

STAT OUT EN is a transparent D latch, open when MBA CLK L is low.

STAT OUT EN <---- DO SLAVE OP H * (SLAVE DONE H + ADD ERROR H)

3.10.2 ST <1:0> L

ST <1:0> L are two open collector transceivers.

ST 1 L is driven low by MCI if STAT OUT EN H * STAT 1 H

ST 0 L is driven low by MCI if STAT OUT EN H * STAT 0 H

3.10.3 RETURN STAT H

RETURN STAT H is a TTL input.

3.11 NR STAT

NR STAT H is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

```
SET NR STAT = (ST <1:Ø> H = Ø) *
                MAST DATA CYC H * DBBZ L

CLR NR STAT = (LD STAT H * IBUS 1 H + INIT H + DT INIT
H)
                * .NOT. SET NR STAT.
```

3.12 CRD

CRD is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

```
SET CRD = (ST <1:Ø> H = 2) * MAST DATA CYC H * DBBZ L.

CLR CRD = (LD STAT H * BUS 29 H + INIT H + DT INIT H.)
                * .NOT. SET CRD
```

3.13 ERR STAT

ERR STAT is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

```
SET ERR STAT = (ST<1:Ø> = 1) * MAST DATA CYC H * DBBZ L

CLR CRD = (LD STAT H * IBUS 3 H + INIT H + DT INIT H) *
                .NOT. SET ERR STAT.
```

3.14 DC LO SYNCH H

DC LO SYNC H is a TTL input.

3.14.1 OLD DC LO SYNC

OLD DC LO SYNC is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

```
OLD DC LO SYNC <---- DC LO SYNC H
```

3.14.2 INIT CTR <1:Ø> H

INIT CTR <1:Ø> H are two edge triggered D flipflops, clocked on the rising edge of MBA CLK L.

```
INIT CTR Ø H <---- (LD CTRL H * IBUS Ø H) + (OLD DC LO
                SYNC H * DC LO SYNC L)

INIT CTR 1 H <---- INIT CTR Ø H
```

3.14.3 INIT L

INIT L is a totem pole output.

INIT H = INIT CTR 0 H + INIT CTR 1 H

3.15 DT FUN <1:0> H

DT FUN <1:0> H are two TTL inputs

They are encoded to define data transfer operations

DT FUN <1:0> H	FUNCTION
0 0	IDLE
0 1	READ
1 0	WRITE
1 1	WCK (writecheck)

3.15.1 DT IDLE

DT IDLE is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

DT IDLE <---- (DT FUN <1:0> H = 0)

3.15.2 DT INIT H

DT INIT H = DT IDLE H * (DT FUN <1:O> 0)

3.16 DT ABORT

DT ABORT is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD DT ABORT = LD CTRL H

DT ABORT <---- IBUS 1 H

CLR DT ABORT = INIT H
A control state to simultaneously load and clear DT ABORT is illegal.

3.16.1 ABORT L

ABORT L is an open collector output. It is driven low by MCI if DT ABORT H + NR STAT H + ERR STAT H + MB EXC H

3.17 MM, MM L

MM is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD MM = LD CTRL H

MM <---- IBUS 3 H

CLR MM = INIT H

A control state to simultaneously load and clear MM is illegal.

MM L is a totem pole output.

3.18 DLY CNTR <4:0>

DLY CNTR <4:0> is a five bit synchronous up counter, clocked on the rising edge of MBA CLK L.

DLY CNTR <4:1> are asynchronously set to ones if DO CMI CYC L.

DLY CNTR <0> is a synchronously set to one if DO DO CMI CYC L * MBA CLK L.

COUNT DLY CNTR <4:0> = (DLY CNTR = 27) * DO CMI CYC H

3.19 EXC SYNC L

EXC SYNC L is a TTL input.

3.19.1 LATCHED EXC SYNC

LATCHED EXC SYNC is a transparent latch, open when MBA CLK L is high.

3.19.2 MB EXC

MB EXC is an edge triggered D flip flop, clocked on the rising edge of MBA CLK L.

SET MB EXC = EXC SYNC H * LATCHED EXC SYNC L

```
CLR MB EXC = (LD STAT * IBUS 7 H + INIT H + DT INIT H)
                  * .NOT. SET MB EXC
```

3.20 IBUS RW H, REG CTRL GA <2:0> H

IBUS RW H and REG CTRL GA <2:0> H are TTL inputs. These signals control the loading and reading of state elements from and to the IBUS signals.

IBUS RW H	REG CTRL GA <2:0> H	Control Signal
0	010	LD VAR H
0	011	LD STAT H
0	100	LD MAINT H
0	101	LD CTRL H
1	0x1	RD STAT
1	100	RD MAINT
1	101	RD CTRL
	OTHER	NOP

3.21 IBUS <29, 28, 16, 7, 3, 2, 1, 0> H

IBUS n H are tristate transeivers,
n = 29, 28, 7, 3, 2, 1, 0.

IBUS 16 H is a tristate driver.

Operation is summarized in the following table for IBUS n H

n	ENABLE	DATA
29	RD STAT H + INIT H	CRD H + INIT H
28	RD MAINT H + INIT H	BLK CMD H + INIT H
16	RD STAT H + INIT H	ATTN H
3	RD STAT H + RD CTRL H = INIT H	RD STAT H * ERR STAT H + RD CTRL H * MM H
2	RD CTRL H + INIT H	IE H + INIT H
1	RD STAT H + RD CTRL H + INIT H	RD STAT H * NR STAT H + RD CTRL H * ABORT H
0	RD CTRL H + INIT H	RD CTRL H * INIT H

MCI Table 1 Pin Identification and Gate Type

PIN #	PIN ID	GATE TYPE
1	CMI 30 H	GA1TTN
2	CMI 31 H	GA1TTN
3	CMI 28 H	GA1TTN
4	CMI 29 H	GA1TTN
5	IBUS 11 H	GA1TZF
6	IBUS 6 H	GA1TZF
7	REG CTRL GA 0 H	GA1TNF
8	DO CMI CYC H	GA1TCF
9	SCLK SYNC L	GA1TNF
10	SILO CTRL MDP 1 H	GA1TTN
11	SILO CTRL MDP 0 H	GA1TTN
14	START STOP MB L	GA1TTF
15	REG CTRL GA 2 H	GA1TNF
16	IBUS RW H	GA1TNF
17	DT DIR H	GA1TNF
18	IBUS 0 H	GA1TZF
19	2ND MBA CLK L	GA1TNF
20	IBUS 1 H	GA1TZF
21	SILO CTRL MDP 2 H	GA1TTN

```
22      SILO CHIP SEL 1 L
        GA1TTN

23      MBA CLK DLY H
                                                  GA1TNF
24      DT FUN 0 H
                                                  GA1TNF
25      EXC SYNC L
                                                  GA1TNF
26      REG CTRL GA 1 H
        GA1TNF

27      LAST TRANS H
                                                  GA1TNF
28      SILO ADDRESS 3 H
        GA1TTN

29      SILO CHIP SEL 0 L
        GA1TTN

30      MBA CLK L
                                                  GA1TNF
31      SILO ADDRESS 4 H
        GA1TTN

32      SILO ADDRESS 2 H
        GA1TTN

33      SILO ADDRESS 1 H
        GA1TTN

34      DT FUN 1 H
                                                  GA1TNF
36      LOAD SILO H
                                                  GA1TTN
37      SILO PAR 0 H
                                                  GA1TZF
39      SILO PAR 1 H
                                                  GA1TZF
40      PAR L
                                                  GA1TNF
41      IBUS 22 H
                                                  GA1TZF
```

```
42    GATED SCLK H
                                             GA1TNF
43    IBUS 31 H
                                             GA1TZF
44    MM L
                                             GA1TNF
45    MB DPA H
                                             GA1TZF
46    IBUS 14 H
                                             GA1TZF
47    ABORT L
                                             GA1TCN
48    CMI BYTE CNTR EQ 1 L
      GA1TNF
```

MCI FIGURE 1

PIN CONFIGURATION DIAGRAM

```
                               ---------------

          CMI 30 H <---!01                   48!<--- CMI BYTE CNT  EQ
1 L
          CMI 31 H <---!02                   47!---> ABORT L
          CMI 28 H <---!03                   46!<--> IBUS 14 H
          CMI 29 H <---!04                   45!<--> MB DPA H
         IBUS 11 H <-->!05                   44!<--- MM L
          IBUS 6 H <-->!06                   43!<--> IBUS 31 H
     REG CTRL GA 0 H  --->!07                42!<--- GATED SCLK H
      DO CMI CYC H <-->!08                   41!<--> IBUS 22 H
       SCLK SYNC L --->!09                   40!<--- PAR L
    SILO CTRL MDP 1 H <---!10 ....... 39!<--> SILO PAR 1 H
    SILO CTRL MDP 0 H <---!11 .     . 38!---- GROUND
                                         VGA ----!12 . LID . 37!<-->
SILO PAR 0 H
                                         VCC ----!13 . DOWN. 36!--->
LOAD SILO H
     START STOP MB L  <-->!14 .     . 35!---- GROUND
     REG CTRL GA 2 H  --->!15 ....... 34!<--- DT FUN 1 H
         IBUS RW H --->!16                   33!---> SILO ADDRESS 1 H
          DT DIR H --->!17                   32!---> SILO ADDRESS 2 H
          IBUS 0 H <-->!18                   31!---> SILO ADDRESS 4 H
     2ND MBA CLK L --->!19                   30!<--- MBA CLK L
          IBUS 1 H <-->!20                   29!---> SILO CHIP SEL 0 L
    SILO CTRL MDP 2 H <---!21                28!---> SILO ADDRESS 3 H
    SILO CHIP SEL 1 L <---!22                27!<--- LAST TRANS H
     MBA CLK DLY H --->!23                   26!<--- REG CTRL GA 1 H
        DT FUN 0 H --->!24                   25!<--- EXC SYNC L

                               ---------------
```

3.14 Massbus Data Path Control (MDC)

1.0 GENERAL DESCRIPTION

MDC is the Massbus Data Path Control Chip. It contains a seven bit up down counter, VAR <8:2>. It controls and maintains status on MAP Party and MAP validity. It detects the beginning and the end of a Data transfer, and maintains status on the success of that transfer. It controls and maintains status on control bus parity. It contains various bits of the control, status and maintenance registers.

2. TERMINOLOGY

Some state elements are described by signals prefixed by SET, LOAD, etc. If none of these are true, the element does not change state.

3. SIGNAL DESCRIPTIONS

3.1 MBA CLK L

MBA CLK L K is a TTL input. It is the primary clock in MDC.

3.2 DT DIR H< DT FUN <1.0> H

These signals are TTL inputs. They describe the direction and the type of data transfer being performed.

DT FUN <1.O)>	Function
00	DT IDLE
01	DT READ
10	DT Write
11	DT WRITECHECK (WCK)

3.2.1 DT IDLE is a transparent latch, open when MBA CLK L is high.

DT IDLE <-- (DT FUN <1:0> H = O>

3.2.2 DT INIT H

DT INIT H = DT IDLE H * (DT FUN < 1:0> H O)

MDC TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	GATE TYPE
1	REG CTRL GA 2 H	GA1TNF
2	REG CTRL GA Ø H	GA1TNF
3	IBUS RW H	GA1TNF
4	IBUS 12 H	GA1TZF
5	IBUS 29 H	GA1TZF
6	MAP PAR L	GA1TNF
7	PAR H	GA1TNF
8	IBUS 3Ø H	GA1TZF
9	XMIT MC PAR H	GA1TTN
1Ø	IBUS 17 H	GA1TZF
11	IBUS 13 H	GA1TZF
14	REC MC PAR H	GA1TNF
15	IBUS Ø H	GA1TZF
16	RUN H	GA1TTN
17	SCLK SYNC L	GA1TNF
18	IBC L	GA1TTN
19	EBL SYNC H	GA1TNF
2Ø	DT BUSY H	GA1TPF
21	DT FUN 1 H	GA1TNF
22	INTER L	GA1TPN
23	OCC SYNC H	GA1TNF
24	SCLK GATE H	GA1TTN
25	DT FUN Ø H	GA1TNF
26	IBUS 31 H	GA1TZF
27	IBUS 6 H	GA1TZF
28	IBUS 2 H	GA1TZF
29	IBUS 7 H	GA1TZF
3Ø	IBUS 3 H	GA1TZF
31	START STOP MB L	GA1TNF
32	MBA CLK L	GA1TNF
33	DT DIR H	GA1TNF
34	UPPER VAR CNT EN L	GA1TTN
36	IBUS 8 H	GA1TZF
37	DT BUSY ONESHOT SYNC H	GA1TNF
39	IBUS 4 H	GA1TZF
4Ø	IBUS 5 H	GA1TZF
41	MAP ERR L	GA1TTN
42	WCK DIF UPR L	GA1TNF
43	WCK DIF LWR L	GA1TNF
44	IBUS 9 H	GA1TZF
45	IBUS 1Ø H	GA1TZF
46	REG CTRL GA 1 H	GA1TNF
47	ABORT L	GA1TPF
48	DO CMI CYC H	GA1TCF

3.3 REG CTRL GA (2:0) H< IBUS RW H.

Reg. CTRL GA (2:0) H and IBUS RW H are TTL INPUTS. They control loading and reading various register bits in MDC. In addition, one code is INIT. Encodings are described in the table below.

	IBUS RW H	REG CTRL GA <2:0> H				
	0				0	
	1			0	R	D
VAR H						
	0	1				
	1			1	L	D
CBDB READ H						
	0			2	L	D
VAR H						
	1	2	RD VAR H			
	0	3	LD STATUS H			
	1			3	R	D
STAT H						
	0			4	L	D
MAINT H						
	1	4	RD MAINT H			
	0	5	LD CTRL H			
	1	5				
	0			6		
	1			6	R	D
MAP H						
	0			7	L	D
BYTE CNT H						
	1			7	INIT	
H						

RD STATUS H = RD STAT H + LD CDBD READ H

3.4 IBUS n H

IBUS n H are 14 TRISTATE Transievers. Operation is summarized in the table below:

N	Enable	Data
0	INIT	0
2	RD VAR H + INIT H	RD VAR H * VAR 2 H + INIT H
3	RD VAR H + INIT H	RD VAR * VAR 3 H
4	RD VAR H + RD STATUS H + INIT H	RD VAR H * VAR 4 H + RD STATUS H * IN V MAP H
5	RD VAR H + RD STATUS H + INIT H	RD VAR H * VAR 5 H + RD STATUS H * MAP PE H
6	RD VAR H + INIT H	RD VAR H * VAR 6 H + INIT H
7	RD VAR H + INIT H	RD VAR H * VAR 7 H + INIT H
8	RD VAR H + RD STATUS H + INIT H	RD VAR H * VAR 8 H + RD STATUS H * MF H
9	RD STATUS H + INIT H	RD STATUS H * WCK ERR LWR H
10	RD STATUS H + INIT H	RD STATUS H * WCK ERR UPR H + INIT H
12	RD STATUS H + INIT H	RD STATUS H * DT ABT H
13	RD STATUS H + INIT M	RD STATUS H * DT COMP H + INIT H
17	RD STATUS H + INIT H	INIT L * MCPE H
29	RD MAINT H + INIT H	INV MAP PAR H + INIT H
30	RD MAINT H + INIT H	INV MCPG H
31	INIT H	INIT H

MDC FIGURE 1

PIN CONFIGURATION DIAGRAM

```
                         ---------------

REG CTRL GA 2 H    --->|01          48|<--> DO CMI CYC H
REG CTRL GA 0 H    --->|02          47|<--> ABORT L
         IBUS RW H --->|03          46|<--- REG CTRL GA 1 H
         IBUS 12 H <-->|04          45|<--> IBUS 10 H
         IBUS 29 H <-->|05          44|<--> IBUS 9 H
         MAP PAR L --->|06          43|<--- WCK DIF LWR L
             PAR H --->|07          42|<--- WCK DIF UPR L
         IBUS 30 H <-->|08          41|---> MAP ERR L
XMIT MC PAR H      <---|09          40|<--> IBUS 5 H
         IBUS 17 H <-->|10 .......  39|<--> IBUS 4 H
         IBUS 13 H <-->|11 .     .  38|---- GROUND
               VGA ----|12 . LID .  37|     <--- DT BUSY ONESHOT SYNC H
               VCC ----|13 . DOWN.  36|<--> IBUS 8 H
      REC MC PAR H --->|14 .     .  35|---- GROUND
          IBUS 0 H <-->|15 .......  34|---> UPPER VAR CNT EN L
             RUN H <---|16          33|<--- DT DIR H
       SCLK SYNC L --->|17          32|<--- MBA CLK L
             IBC L <---|18          31|<--- START STOP MB L
        EBL SYNC H --->|19          30|<--> IBUS 3 H
         DT BUSY H <-->|20          29|<--> IBUS 7 H
        DT FUN 1 H --->|21          28|<--> IBUS 2 H
          INTER L  <---|22          27|<--> IBUS 6 H
        OCC SYNC H --->|23          26|<--> IBUS 31 H
       SCLK GATE H <---|24          25|<--- DT FUN 0 H
                     ---------------
```

3.5 VAR <8:2> H

VAR <8:2> H is a seven bit up/down synchronous counter. It is clocked on the rising edge of MBA CLK L.

If LD VAR H, VAR <8:2> H <-- IBUS <8:2> H

COUNT UP VAR = DT DIR H * INC VAR H
COUNT DOWN VAR = DT DIR L * INC VAR H

Where

INC VAR H = DOING CMI CYCLE H * INV MAP L * MAP PE L * DO CMC CYC L

3.5.1 UPPER VAR CNT EN L

UPPER VAR CNT EN L is a totempole output.
UPPER VAR CNT EN L = [(VAR <8:2 H = 7F) * DT DIR H + (VAR <8:2> = 0) * DT DIR L] * INC VAR H

3.6 UPDATE MAP

UPDATE MAP is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

SET UPDATE MAP = UPPER VAR CNT EN H + DT INIT H
CLR UPDATE MAP = (RD MAP H + INIT H)* .NOT.SET UPDATE MAP

3.6.1 INV MAP PEND

INV MAP PEND is a transparent latch, open if UPDATE MAP H *
MBA CLK H.

INV MAP PEND <-- IBUS 31 L.

3.6.2 INV MAP

INV MAP is an edge triggered flipflop, clocked on the rising edge of MBA CLK L. It is asychronously set if INV MAP PEND H * DO CMI CYC H.

CLR INV MAP = DT INIT H + INIT H + LD STATUS H * IBUS 4 H

3.6.3 INV MAP PAR

INV MAP PAR H is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

LOAD INV MAP PAR = LD MAINT H

INV MAP PAR <-- IBUS 29 H

CLR INV MAP PAR = INIT H * .NOT. LOAD INV MAP PAR

3.6.4 MAP PAR L

MAP PAR L is a TTL input.

3.6.5 MAP PE PEND

MAP PE PEND is a transparent latch, open if UPDATE MAP H * MBA CLK H

MAP PE PEND <-- MAP PAR H @ INV MAP PAR L

3.6.6 MAP PE

MAP PE is an edge triggered flipflop, clocked on the rising edge of MBA CLK L. It is asynchronously set if MAP PE PEND H * DO CMI CYC H.

CLR MAP PE = INIT H + DT INIT H + LD STATUS H * IBUS 5 H

3.6.7 MAP ERR L

MAP ERR L is a totempole output.

MAP ERR H = MAP PE H + INV MAP H

3.6.8 DO CMI CYC H

DO CMI CYC H is an open collector transiever. It is driven low by MDC if UPDATE MAP H + INV MAP H + MAP PE H.

3.6.9 DOING CMI CYC

DOING CMI CYC is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

DOING CMI CYC <-- DO CMI CYC H

3.7 EBL SYNC H

EBL SYNC H is a TTL input

3.7.1 LATCHED EBL SYNC

LATCHED EBL SYNC is a transparent D Latch, open when MBA CLK L is low.

LATCHED EBL SYNC <-- EBL SYNC H.

3.7.2 PREV EBL SYNC

PREV EBCL SYNC is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

PREV EBL SYNC <--- EBL SYNC H.

3.7.3 MB DONE

MB DONE is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

SET MB DONE = PREV EBL SYNC H * EBL SYNC L * RUN L + MXF 4* DT IDLE L

CLR MB DONE = DT IDLE H * .NOT. SBT MB DONE

3.7.4 DT BUSY H

DT BUSY H is an open collector transiever. It is driven low by MDC if MB DONE H * DT COMP H.

3.8 DT ABT

DT ABT is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

SET DT ABT = MB DONE H * ABORT H

CLR DT ABT = (INIT H + DT INIT H + LD STATUS * IBUS 12 H) * .NOT. SET DT ABT.

3.9 LATCHED ABORT

LATCHED ABORT is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

LATCHED ABORT <-- ABORT H

3.9.1 PRERUN ABORT

PRERUN ABORT is an edge triggered flipflop, clocked with the rising edge of MBA CLK L.

SET PRERUN ABORT = LATCHED ABORT L * ABORT H * RUN L * PRERUN EN H.

CLR PRERUN ABORT = (INIT H + DT INIT H + EBL SYNC L + MXF L) * .NOT. SET PRERUN ABORT.

3.10 START STOP MB L

START STOP MB L is a TTL Input.

3.10.1 SCLK GATE

SCLK GATE is an edge triggered flipflop, clocked cn the rising edge of MBA CLK L.

SET SCLK GATE = START STOP MB H * DT FUN 1 H * ABORT L + DT INIT H * DT READ H

CLR SCLK GATE = (INIT H + ABORT H + END MB DT H) * .NOT. SET SCLK GATE

Where

END MB DTH = START STOP MB H * DT READ H * SCLK SYNC H + START STOP MB L * DT FUN 1 H * SCLK SYNC H + EBL SYNC H * IBC H.

3.10.2 DT RUN

DT RUN is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

SET DT RUN = DT INIT H * DT READ H + DT FUN 1 H * SCLK GATE H * LATCHED ABORT L * END MB DT L.

CLR DT RUN = (MXF H + LATCHED ABORT H + END MB DT H + INIT H) * .NOT. SET DT RUN.

3.10.3 PRE-RUN EN

PRE-RUN EN is an SR Latch.

SET PRE-RUN EN = MBA CLK H * DT RUN H.

CLR PRE-RUN EN = MBA CLK H * ENABLE TIMEOUT H

3.10.4 RUN H

RUN H is totempole output.

RUN H = DT RUN H + PRERUN ABORT H

3.11 WCK DIF UPR L, WCK DIF LWR L

WCK DIF UPR L AND WCK DIF LWR L are TTL inputs.

3.11.1 WCK ERR UPR H

WCK ERR UPR H is an edge triggered flipflop, clocked on the rising edge of MBA CLK C.

SET WCK ERR UPR = START STOP MB H + ODD LENGTH DT L + DT DIR L) * WCK DIF UPR H * WCK COMP EN H

Where

WCK COMP EN H = SCLK SYNC H * PRERUN ABORT L * RUN H * DT WCK H

CLR WCK ERR UPR = (INIT H + DT INIT H + LD STATUS H * IBUS 10H) * .NOT. SET WCK UPR

3.11.2 WCK ERR LWR

WCK ERR LWR is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

SET WCK ERR LWR = START STOP MB H + ODD LENGTH DT L + DT DIR H) * WCK DIF LWR H * WCK COMP EN H.

CLR WCK ERR LWR = (INIT H + DT INIT H + LD STATUS H * IBUS 9 H) * .NOT SET WCK ERR LWR.

3.12 OCC SYNC H, SCLK SYNC H, DT BUSY ONESHOT SYNC H.

These signals are TTL inputs.

3.12.1 ENABLE TIMEOUT

ENABLE TIMEOUT is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

SET ENABLE TIMEOUT = DT BUSY H * DT IDLE H

CLR ENABLE TIMEOUT = (DT IDLE H + OCC SYNC H) * .NOT. SET ENABLE TIMEOUT

3.12.2 MXF

MXF is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

SET MXF = DT Busy oneshot SYNCL * ENABLE TIMEOUT H

CLR MXF = (INIT H + DT INIT H + LD STATUS H * IBUS 8 H) * .NOT. SET MXF

3.13 INTER L

INTER L is an open collector output. MDC drives INTER L low if
(MXF H + MCPE H + DT COMP H + DT ABT H) * DT BUSY L.

3.14 IBC, IBC L

IBC is an edge triggered flipflop clocked on the rising edge of MBA CLK L. IBC L is a totempole output.

LOAD IBC = LD CTRL H

IBC <-- IBUS 4 H

CLR IBC = INIT H * .NOT. LOAD IBC

3.15 PAR H, REC MC PAP H

These signals are TTL inputs.

3.15.1 INV MCPG

INV MCPB is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

LOAD INV MCPG = LD MAINT H

INV MCPG <-- IBUS 30 H

CLR INV MCPG = INIT H * .NOT. LOAD INV MCPG

3.15.2 XMIT MC PAR H

XMIT MC PAR H is a totempole output.

XMIT MC PAR H = PAR H ⊕ INV MCPG H

3.15.3 MCPE

MCPE is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

SET MCPE = LD CBDB H * INIT L * (PAR H + REC MC PAR L)

CLR MCPE = (INIT H + LD STATUS H * IBUS 17 H) * .NOT. SET MCPE

3.15 MASSBUS DATA PATH (MDP) DC645

1. **GENERAL DESCRIPTION:**

This specification defines the detail requirements for the VAX-11/750 MDP gate array chip.

MDP TABLE 1

Pin Identification and Gate Type

			GATE				GATE
PIN #	PIN ID	COMMENTS	TYPE	PIN #	PIN ID	COMMENTS	TYPE
1	DT FUN 0 H		GA1TNF	25	ADD COMP 1 H		GA1TPF
2	MB 0 H		GA1TZF	26	ADD COMP 2 H		GA1TPF
3	GATED SCLK H		GA1TNF	27	SCLK SYNC L		GA1TNF
4	MB 1 H		GA1TZF	28	MB BYTE CNTR EQ 1		GA1TCN
5	SLAVE READ L		GA1TNF	29	MB CRY IN H		GA1TNF
6	MM L		GA1TNF	30	MB CRY OUT H		GA1TTN
7	WCK ERR L		GA1TCN	31	CMI CRY IN L		GA1TNF
8	PAR H		GA1TTN	32	CMI CRY OUT L		GA1TTN
9	MBA CLK L		GA1TNF	33	REG CTRL MDP 2		GA1TNF
10	CMI 0 H		GA1TZF	34	CMI BYTE CNTR EQ 1		GA1TCN
11	CMI 1 H		GA1TZF	36	MBA INIT L		GA1TNF
14	CB 0 H		GA1TZF	37	CMI 17 H		GA1TZF
15	CB 1 H		GA1TZF	39	CMI 16 H		GA1TZF
16	IBUS 16 H		GA1TZF	40	VECTOR CYC L		GA1TNF
17	IBUS 17 H		GA1TZF	41	SILO CTRL MDP 1		GA1TNF
18	IBUS 0 H		GA1TZF	42	MBA DBBZ H		GA1TNF
19	TRA SYNC L		GA1TNF	43	DO CMI MASTER L		GA1TNF
20	ADD COMP 0 H		GA1TPF	44	SILO CTRL MDP 0		GA1TNF
21	IBUS 1 H		GA1TZF	45	DT FUN 1 H		GA1TNF
22	REG CTRL MDP 1		GA1TNF	46	SILO CTRL MDP 2		GA1TNF
23	REG CTRL MDP 0		GA1TNF	47	SILO 0 H		GA1TZF
24	IBUS RW H		GA1TNF	48	SILO 1 H		GA1TZF

2. Performance Requirements:

2.1 GENERAL

MDP is the MBA data path chip. It routes data between 5 tristate ports. It supplies one parity output used for a variety of registers and ports dependent on control signals. One output provides equals checking on two of the data registers. Two 2-bit counters are included. Logic is included for address comparison to the CMI port. 3 bits of ID are latched internally at initialization.

2.1.1 TERMINOLOGY

The next state of state elements is sometimes described by signals such as "SET xxx", "CLR xxx", and "TOGGLE xxx". If these signals are true, that element is set, cleared or toggled respectively. If none are true, the element does not change state.

2.2 DATA PORTS

All data ports are high asserted and tristate.

1. CMI - 4 bits - 17,16,1,0
2. IBUS - 4 bits - 17,16,1,0
3. SILO - 2 bits - 1,0
4. MB - 2 bits - 1,0
5. CB - 2 bits - 1,0

2.3 CLOCKS

Two clocks are input to MDP. MBA CLK L is the primary clock. Most of the state of MDP is latched with the low to high transition of MBA CLK L.

GATED SCLK H is used for clocking the registers on the input and output of the MB port. Clocking is done with the high to low transition of GATED SCLK H.

2.4 INIT L

The assertion of INIT L initializes some of the state elements of MDP. Initialization is fully described in later sections for each of these state elements.

2.5 CONTROL FIELDS

There are three major control fields. Operation is summarized in the tables below, and fully described in the later sections.

2.5.1 DT FUN <1:0> H

00	-	IDLE
01	-	DT READ
10	-	DT WRITE
11	-	DT WCK

2.5.2 IBUS RW H & REG CTRL MDP <2:0> H

0-0	NOP
0-1	IBUS <- CMI IBUF
0-2	RESERVED
0-3	IBUS <- CMI IBUF
0-4	CMI OBUF <- Byte Counters
0-5	Byte Counters <- CMI IBUF, IBUS <- CMI IBUF
0-6	CMI OBUF<- CMD ADD REG
0-7	CBDB <- CMI IBUF, IBUS <- CMI IBUF
1-0	CMI OBUF <- IBUS
1-1	CMI OBUF <- IBUS
1-2	CMI OBUF <- IBUS
1-3	CMD ADD REG <- IBUS
1-4	CMI OBUF <- IBUS & MDIB
1-5	CMI OBUF <- IBUS
1-6	CMI OBUF <- IBUS & MDIB
1-7	CMI OBUF <- IBUS & CBDB

Operation is highly dependent on ID.

2.5.3 SILO CTRL MDP <2:0) H

This field is used in conjunction with DT FUN 1 H.

DT FUN 1 H - SILO CTRL MDP

0-0	NOP
0-1	SILO <- SMDB
0-2	RESERVED
0-3	SMDB <- MDIB, Gen PAR
0-4	SCDB LO <- SILO, Gen PAR
0-5	SCDB LO <- SILO, Gen PAR
0-6	SCDB HI <- SILO, Gen PAR
0-7	SCDB HI <- SILO, Gen PAR
1-0	NOP
1-1	SILO <- 0
1-2	Gen PAR on MDIB
1-3	SMDB <- SILO, Gen PAR
1-4	SILO <- SCDB LO, Gen PAR
1-5	SILO <- SCDB LO, Gen PAR
1-6	SILO <- SCDB HI, Gen PAR
1-7	SILO <- SCDB HI, Gen PAR

2.6 MM L

When MM L is asserted, the drivers of CB <1:0> H and MB <1:0> H are enabled.

2.7 ID

MDP contains three bits of ID. These three bits are held internally in latches. ID <2:0> is latched with the data on IBUS <17,1,0>. The latches are open when MBA CLK L is low and INIT L is low.

2.8 CMI IBUF

CMI IBUF is a four bit transparent latch. It is open during the low pulse of MBA CLK L if DATA CYC H is high. CMI IBUF n H is loaded with CMI n H respectively, where n = 0,1,16,17.

2.9 CMI OUTPUT MUX & CMI OBUF

2.9.1 The CMI OUTPUT MUX steers command/address, data, and vector into the CMI OBUF. The CMI OBUF is a transparent latch which is open during the low pulse of MBA CLK L if LATCH CMI OBUF EN is asserted.

LATCH CMI OBUF EN = SLAVE READ H * DATA CYC H + CMI CMD EN H + MASTER DATA CYC H.

The CMI OUTPUT MUX is controlled by REG CTRL MDP <2:0> H, IBUS RW H, VECTOR CYC H, CMI CMD EN H, and ID as shown in Table 3.3.9.

VECTOR OUT EN = VECTOR CYC H * MASTER DATA CYC H
SCDB OUT EN = VECTOR CYC L * MASTER DATA CYC H *DT FUN 1 L * DT FUN 0 H

2.9.2 OUTPUT ENABLES

The CMI OBUF is asserted on the CMI through tri-state drivers when enabled. The output enables are transparent latches, open when MBA CLK L is low. CMI <1:0> and CMI <17:16> are enabled independently.

SET CMI 1:0 OUT EN = CMI CMD EN H + SLAVE READ H

CLR CMI 1:0 OUT EN = (DATA CYC L + DT FUN 1 H) * .NOT. SET CMI 1:0 OUT EN

SET CMI 17:16 OUT EN = CMI CMD EN H * (ID $\leq$ 5) + SLAVE READ H

CLR CMI 17:16 OUT EN = (DATA CYC L + DT FUN 1 H) * .NOT. SET CMI 17:16 OUT EN

The output enables are cleared asynchronously by INIT.

2.10 CMI CONTROL STATE

There are four edge triggered flip flops that are used to control putting control/data onto and receiving control/data off the CMI. They are clocked on the positive going edge of MBA CLK L.

2.10.1 ADD CYC POS - DATA CYC

ADD CYC POS <- MBA DBBZ L
DATA CYC = .NOT. ADD CYC POS

2.10.2 CMI CMD EN

CMI CMD EN <- MBA DBBZ L * INIT L * DO CMI MASTER H

2.10.3 MASTER DATA CYC

MASTER DATA CYC <- CMI CMD EN H *INIT L* (DT FUN 1 L + VECTOR CYC H)

2.10.4 RD DATA PEND

SET RD DATA PEND = CMI CMD EN H * DT FUN 1 H
CLR RD DATA PEND = (MBA DBBZ L + INIT) *.NOT. SET RD DATA PEND

2.11 BYTE COUNTERS

2.11.1 MDP contains 2 two-bit counters - the MB BYTE COUNTER and the CMI BYTE COUNTER. They are loaded simultaneously, but incremented separately. The following signal pins are associated with the BYTE COUNTERS.

SIGNAL PIN	DRIVE/REC	ASSOCIATED COUNTER
MB CRY IN H	R	MB BYTE COUNTER
MB CRY OUT H	D(TT)	MB BYTE COUNTER
MB BYTE CNTR EQ -1 H	D(OC)	MB BYTE COUNTER
SCLK SYNC L	R	MB BYTE COUNTER
CMI CRY IN L	R	CMI BYTE COUNTER
CMI CRY OUT L	D(TT)	CMI BYTE COUNTER
CMI BYTE CNTR EQ -1 H	D(OC)	CMI BYTE COUNTER

All transitions in state of the BYTE counters occur on the positive going edge of MBA CLK L.

2.11.2 LOADING COUNTERS

The counters are loaded from the CMI IBUF if (REG CTRL MDP = 5) * (IBUS RW = 0)

```
MB  BYTE CNTR 0 <- CMI IBUF 0
MB  BYTE CNTR 1 <- CMI IBUF 1
CMI BYTE CNTR 0 <- CMI IBUF 0
CMI BYTE CNTR 1 <- CMI IBUF 1
```

2.11.3 INCREMENTING COUNTERS

```
TOGGLE MB BYTE CNTR 0 = MB CRY IN H * (1D ≠ 0) * SCLK
SYNC H
TOGGLE MB BYTE CNTR 1 = [(MB CRY IN H * MB BYTE CNTR 0
H)
       + ID = 0] * S CLK SYNC H.
TOGGLE CMI BYTE CNTR 0 = CMI CRY IN H * SILO CTRL MDP 2
H
TOGGLE CMI BYTE CNTR 1 = CMI CRY IN H * CMI BYTE CNTR 0
H
       *SILO CTRL MDP 2 H.
CLR MB BYTE CNTR 0 = (MB CRY IN H * (1D = 0) * SCLK SYNC
H
```

2.11.4 RESTRICTION

RESTRICTION - Control states that increment and load simultaneously are illegal, and lead to unpredictable results.

2.11.5 COUNTER OUTPUTS

```
MB CRY OUT H = (MB BYTE CNTR 0 H + (ID = 0)) *
MB BYTE CNTR 1 H * (MB CRY IN H + (ID = 0))

MB BYTE CNTR EQ -1 H = (MB BYTE CNTR 0 H + (ID = 0)) *
      MB BYTE CNTR 1 H

CMI CRY OUT H = CMI BYTE CNTR 1 H *
      CMI BYTE CNTR 0 H * CMI CRY IN H

CMI BYTE CNTR EQ -1 H = CMI BYTE CNTR 1 H *
      CMI BYTE CNTR 0 H
```

2.12 CB n H & CBDB

CBDB is an edge triggered 2 bit buffer. It can drive or be loaded from CB n H. Additionally it can be loaded from CMI IBUF. It is clocked on the rising edge of MBA CLK L.

```
CBDB <- CMI IBUF if (REG CTRL MDP = 7) * IBUS RW L
CBDB <- CB n H   if TRA SYNC H * SLAVE READ H
```

CB n H are tri-state signals, enabled (CB OUT EN H + MM H)

CB OUT EN is an edge triggered flop clocked by the rising edge of MBA CLK L.

```
SET CB OUT EN = (REG CTRL MDP = 7) * IBUS RW L
CLR CB OUT EN = TRA SYNC H * .NOT. [(REG CTRL MDP = 7)
      *IBUS RW L]
```

CB OUT EN is asynchronously cleared by INIT.

2.13 IBUS n H

IBUS n H (n = 17,16,1,0) is a tri-state bidirectional port. MDP asserts the CMI IBUF onto the IBUS when IBUS RW L * REG CTRL MDP 0 H.

2.14 SCDB

SCDB is a four bit edge triggered register clocked with the rising edge of MBA CLK L. It is treated as one register when loading from or reading to the CMI. It is treated as two registers, SCDB LO = SCDB <1:0> & SCDB HI = SCDB <17:16>, when loading from or reading to the SILO.

```
                    SCDB <- CMI IBUF if RD DATA PEND H = 1
SCDB LO <- SILO <1:0> H if SILO SCDB EN * SILO CTRL MDP 1 L * DT FUN 1 L
SCDB HI <- SILO <1:0> H if SILO SCDB EN * SILO CTRL MDP 1 H * DT FUN 1 L
where SILO SCDB EN = SILO CTRL MDP 2 H * (SILO CTRL MDP 0 H + ID 2 L)
```

RD DATA PEND H and SILO SCDB EN being both asserted is an illegal control state.

2.15 SILO <1:0> H

SILO <1:0> H is a bidirectional tri-state port. It is enabled if:

```
(SILO CTRL MDP <2:0> = 1) + (DT FUN 1 H * SILO SCDB EN)
```

When enabled,

```
SILO <1:0> H <- SCDB HI if
   SILO CTRL MDP 2 H * SILO CTRL MDP 1 H

SILO <1:0> H <- SCDB LO if
   SILO CTRL MDP 2 H * SILO CTRL MDP 1 L

SILO <1:0> H <- SMDB if DT FUN 1 L *
   (SILO CTRL MDP <2:0> = 1)

SILO <1:0> H <- 0 if DT FUN 1 H *
```

(SILO CTRL MDP <2:0> = 1)

2.16 SMDB

SMDB is a 2 bit edge triggered register clocked with the rising edge of MBA CLK L.

SMDB <- MDIB if (SILO CTRL MDP =3) * DT FUN 1 L
SMDB <- SILO <1:0> H if (SILO CTRL MDP = 3) * DT FUN 1 H

2.17 MDIB

MDIB <1:0> is a 2 bit edge triggered register clocked with the falling edge of GATED SCLK H. The input is MB <1:0> H.

2.18 MDOB

MDOB <1:0> is a 2 bit edge triggered register clocked with the falling edge of GATED SCLK H. The input is SMDB. < 1:0 > H

2.19 MB <1:0> H

MB <1:0> H is a bidirectional tri-state port. The drivers are enabled if (DT FUN 1 H * DT FUN 0 L) + MM H. When enabled, MB <1:0> H <- MDOB <1:0> H.

2.20 WCK ERR L

WCK ERR L is an open collector signal.
WCK ERR L is asserted when MDIB <1:0> ≠ SMDB <1:0>

2.21 PAR H

PAR H is an totempole signal which is equivalent to odd parity generation. The data selected is shown in the table below.

CONTROL FIELDS	DATA SIGNALS
(SILO CTRL MDP <2:0> = 3) * DT FUN 1 H	SILO <1:0> H
(SILO CTRL MDP <2:0> = 3) * DT FUN 1 L	MDIB <1:0> H
(SILO CTRL MDP <2:0> = 2) * DT FUN 1 H	MDIB <1:0> H
DT FUN 1 H * SILO SCDB EN * SILO CTRL MDP 1 L	SCDB <1:0> H
DT FUN 1 H * SILO SCDB EN * SILO CTRL MDP 1 H	SCDB <17:16> H
DT FUN 1 L * SILO SCDB EN	SILO <1:0> H

where SILO SCDB EN = SILO CTRL MDP 2 H * (SILO CTRL MDP 0 H ⊕ ID 2 L)

In any control state other than those listed, PAR H <- 0.

2.22 ADD COMP <2:0> H

ADD COMP <2:0> H are bidirectional signals used for comparing CMI addresses, latching part of the CMI address, and for inputing the vector value to MDP.

Operation is dependent on ID, VECTOR CYC L, ADD CYC POS. See MDP Table 3.

ADD COMP <2:0> H are open collector drivers.

In MDP TABLE 3, OUTPUT implies that the value of the output is latched. The latches are open when MBA CLK L is low if ADD CYC POS H is high. Note that the latch can be latched low if the output is being pulled low by another driver, even if it would otherwise be latched high. INPUT implies that the driver is disabled.

2.23 CMD/ADD REG

CMD/ADD REG is a four bit transparent register. CMD/ADD <17:16, 1:0> is loaded from IBUS <17:16, 1:0>. It is open when MBA CLK L is low if (REG CTRL MDP =3) * IBUS RW H.

2.24 I/O SIGNALS

SIGNAL NAME		# OF PINS	TYPE OF I/O
ADD COMP n H	0 ≤ n ≤ 2	3	XCEIVER-OC
IBUS n H	n = 17,16,1,0	4	XCEIVER-3 STATE
CB n H	n = 0,1	2	XCEIVER-3 STATE
SCLK SYNC L		1	RECEIVER
MB CRY IN H		1	RECEIVER
MB CRY OUT H		1	DRIVER-TT
MB BYTE CNTR EQ -1 H		1	DRIVER-OC
CMI CRY IN L		1	RECEIVER
CMI CRY OUT L		1	DRIVER-TT
CMI BYTE CNTR EQ -1 H		1	DRIVER-OC
SILO CTRL MDP n H	0≤n≤2	3	RECEIVER
SILO n H	n= 0,1	2	XCEIVER-3 STATE
GATED SCLK H		1	RECEIVER
MB n H	n = 0,1	2	XCEIVER-3 STATE
PAR H		1	DRIVER-TT
WCK ERR L		1	DRIVER-OC
REG CTRL MDP n H	0≤n≤2	3	RECEIVER
IBUS RW H		1	RECEIVER
CMI n H	n = 17,16,1,0	4	XCEIV-3 STATE
SLAVE READ L		1	RECEIVER
INIT L		1	RECEIVER
DT FUN n H	n = 1,0	2	RECEIVER
MM L		1	RECEIVER
MBA CLK L		1	RECEIVER
TRA SYNC L		1	RECEIVER
MBA DBBZ H		1	RECEIVER
DO CMI MASTER L		1	RECEIVER
VECTOR CYC L		1	RECEIVER
TOTAL		44	

```
                          ---------------
      DT FUN 0 H --->!01           48!<--> SILO 1 H
          MB 0 H <-->!02           47!<--> SILO 0 H
    GATED SCLK H --->!03           46!<--- SILO CTRL MDP 2 H
          MB 1 H <-->!04           45!<--- DT FUN 1 H
    SLAVE READ L --->!05           44!<--- SILO CTRL MDP 0 H
            MM L --->!06           43!<--- DO CMI MASTER L
       WCK ERR L <---!07           42!<--- MBA DBBZ H
           PAR H <---!08           41!<--- SILO CTRL MDP 1 H
       MBA CLK L --->!09           40!<--- VECTOR CYC L
        CMI 0 H  <-->!10 ....... 39!<--> CMI 16 H
        CMI 1 H  <-->!11 .     . 38!---- GROUND
             VGA ----!12 . LID . 37!<--> CMI 17 H
             VCC ----!13 . DOWN. 36!<--- MBA INIT L
          CB 0 H <-->!14 .     . 35!---- GROUND
          CB 1 H <-->!15 ....... 34!---> CMI BYTE CNTR EQ 1 L
      IBUS 16 H <-->!16           33!<--- REG CTRL MDP 2 H
      IBUS 17 H <-->!17           32!---> CMI CRY OUT L
        IBUS 0 H <-->!18           31!<--- CMI CRY IN L
     TRA SYNC L --->!19           30!---> MB CRY OUT H
  ADD COMP 0 H <-->!20           29!<--- MB CRY IN H
        IBUS 1 H <-->!21           28!---> MB BYTE CNTR EQ 1 L
REG CTRL MDP 1 H --->!22           27!<--- SCLK SYNC L
REG CTRL MDP 0 H --->!23           26!<--> ADD COMP 2 H
       IBUS RW H --->!24           25!<--> ADD COMP 1 H
                          ---------------
```

MDP FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

MDP TABLE 2
MDP CHIP SPEC

IBUS RW H	REG CTRL MDP	VECTOR OUT EN H	CMI CMD EN H	1D	SCDB OUT EN H	OPERATION
X*	X*	0	1	X	0	CMI OMUX <17:16, 1:0> <- CMD/ADD REG <17:16, 1:0>
X*	X*	1	0	>4	0	CMI OMUX <17:16, 1:0> <- 0
X*	X*	1	0	=4	0	CMI OMUX <17:16, 1> <- 0, CMI OMUX <0> <- ADD COMP 0 H
X*	X*	1	0	≤3	0	CMI OMUX <17:16> <- 0, CMI OMUX <1:0> <- ADD COMP <1:0> H
0	4	0	0	X	0	CMI OMUX <17:16> <- MB BYTE CNTR <1:0>, CMI OMUX <1:0> <- CMI BYTE CNTR <110>
0	6	0	0	X	0	CMI OMUX <17:16> <- CMD/ADD REG <17:16, 1:0>
0	0THER	0	0	X	0	CMI OMUX <17:16, 1:0> <- 0
1	0	0	0	X	0	CMI OMUX <17:16, 1:0> <- IBUS <17:16, 1:0>
1	1	0	0	<6	0	CMI OMUX <17:16> <- 0, CMI OMUX <1:0> <- IBUS <1:0>
1	1	0	0	7,6	0	CMI OMUX <17,0>,IBUS <17.0> CMI OMUX <16,1> <- 0
1	2	0	0	≤3	0	CMI OMUX <17:16> <- 0, CMI OMUX <1:0> <- IBUS <1:0>
1	2	0	0	≥4	0	CMI OMUX <17:16, 1:0> <- 0
1	4	0	0	≤3	0	CMI OMUX <17:16> <- IBUS <17:16>, CMI OMUX <1:0> <- MDIB <1:0>
1	4	0	0	>4	0	CMI OMUX <17:16, 1:0> <- IBUS <17:16, 1:0>
1	5	0	0	≤1	0	CMI OMUX <17:16,1> <- IBUS <17:16,1>, CMI OMUX <0> <- 0
1	5	0	0	2:6	0	CMI OMUX <17,1:0> <- IBUS <17,1:0>, CMI OMUX <16> <- 0
1	5	0	0	7	0	CMI OMUX <17,0>,IBUS <17.0> CMI OMUX <16,1> <- 0

MDP TABLE 2 (Cont)
MDP CHIP SPEC

		VECTOR	CMI		SCDB	
IBUS	REG	OUT	CMD		OUT	
RW	CTRL	EN	EN		EN	
H	MDP	H	H	1D	H	OPERATION
1	6	0	0	X	0	CMI OMUX <17:16> <- IBUS <17:16>, CMI OMUX <1:0> <- MDIB <1:0>
1		0	0	$\leq$1	0	CMI OMUX <17:16> <- IBUS <17:16>, CMI OMUX <1:0> <- CBDB <1:0>
1		0	0	3:5	0	CMI OMUX <17:16> <- 0, CMI OMUX <1:0> <- CBDB <1:0>
1		0	0	2,6,7		0 CMI OMUX <17> <- IBUS <17>, CMI OMUX <16> <- 0, CMI OMUX <1:0> <- CBDB <1:0>
1	OTHER	0	0	X	0	UNDEFINED
X*	X*	0	0	X	1	CMI OMUX <17:16, 1:0> <- SCDB <17:16, 1:0>

* It is not possible to have more than on of VECTOR OUT EN,CMI CMD EN) or SCBD EN asserted at one time. If one of these three signals is asserted, IBUS RW H and REG CTRL MDP must be 0-0, 0-1, 0-2, 0-3, 0-5, 0-7 or 1-3. Any other combination leads to unpredictable results.

MDP TABLE 3
MDP CHIP SPEC

ID	ADD CYC POS	VECTOR CYC #	ADD COMP 2	ADD COMP 1	ADD COMP Ø
Ø	1	Ø	OUTPUT CMI< 17 H * CMI 16 L	OUTPUT <- CMI 1 H	OUTPUT <- CMI O H
1	1	Ø	OUTPUT <-CMI 17 L * CMI 16 L	OUTPUT <- CMI 1 H	OUTPUT <-CMI Ø H
2	1	Ø	OUTPUT <- CMI 17 H * CMI 16 H	OUTPUT <-CMI 1 H	OUTPUT <- CMI O H
3	1	Ø	OUTPUT <- CMI 17 H * CMI 16 H	OUTPUT <- CMI 1 H	OUTPUT <- CMI Ø H
4	1	Ø	OUTPUT <- CMI 17 H	OUTPUT <- CMI 1 H	OUTPUT <- CMI Ø H
5	1	Ø	OUTPUT <- CMI 17 H	OUTPUT <- CMI 1 H	OUTPUT <- CMI Ø H
6	1	Ø	OUTPUT <- CMI 17 H * CMI 16 H	OUTPUT <- CMI Ø L * (CMI 1 H = ADD COMP Ø H)	INPUT
7	1	Ø	OUTPUT <- CMI 17 H * CMI 16 H	OUTPUT <- CMI 1 H * (CMI Ø H = ADD COMP Ø H)	INPUT
*	Ø	1	OUTPUT <- LATCHED VALUE	INPUT	INPUT
*	Ø	Ø	OUTPUT <- LATCHED VALUE	OUTPUT <- LATCHED VALUE	OUTPUT <- LATCHED VALUE

3.16 MEMORY DATA REGISTER (MDR)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for a Memory Data Register Chip (MDR). The MDR chip is a 4 bit slice which accomplishes data routing and alignment functions, on one bit per byte of data, between the main CPU data paths and the memory interface.

MDR TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	CLK SEL S1 H		GA1TNG
2	CLK SEL S0 H		GA1TNG
3	B CLK L		GA1TNG
4	ADD REG ENA L		GA1TNG
5	ENA CMI L		GA1TNG
6	PC 00 H		GA1TNG
7	PC 01 H		GA1TNG
8	XB SELECT H		GA1TNG
9	XBUF 00 H		GA1TTN
10	SNAPSHOT CMI L		GA1TNG
11	XBUF 08 H		GA1TZG
14	DBUS ROT S0 H		GA1TNG
15	DBUS ROT S1 H		GA1TNG
16	CM 24 H		GA1TZG
17	CM 16 H		GA1TZG
18	CM 08 H		GA1TZG
19	CM 00 H		GA1TZG
20	WB 00 H		GA1TNI
21	WB 08 H		GA1TNI
22	WB 16 H		GA1TNI
23	WB 24 H		GA1TNI
24	CACHE 24 H		GA1TZI
25	CACHE 16 H		GA1TZI
26	CACHE 08 H		GA1TZI
27	CACHE 00 H		GA1TZI
28	AMUX SEL S0 H		GA1TNG
29	AMUX SEL S1 H		GA1TNG
30	PAD 00 H		GA1TZG
31	PAD 08 H		GA1TZG
32	PAD 16 H		GA1TZG
33	DBUS SEL S0 H		GA1TNG
34	DBUS SEL S1 H		GA1TNG
36	CHIP ID H		GA1TNG
37	MAD 00 H		GA1TNG
39	MAD 16 H		GA1TNG
40	MAD 08 H		GA1TNG
41	MMUX SEL S0 H		GA1TNG
42	MB 00 L		GA1TZN
43	MMUX SEL S1 H		GA1TNG
44	MB 08 L		GA1TZN
45	MAD 24 H		GA1TNG
46	MB 16 L		GA1TZN
47	MB 24 L		GA1TZN
48	MBUS ENA H		GA1TNF

2 **Performance:** The MDR Chip shall interface to the following busses:

A. Input Busses:

WBUS	(WB 24 H, WB 16 H, WB Ø8 H, WB ØØ H)
MA BUS	(MAD 24 H, MAD 16 H, MAD Ø8 H, MAD ØØ H)

B. Output Bus:

MBUS	(-MB 24 H, -MB 16 H, -MB Ø8 H, -MB ØØ H)

C. Bidirectional Busses:

PA BUS	(PAD 16 H, PAD Ø8 H, PAD ØØ H)
CACHE BUS	(CACHE 24 H, CACHE 16 H, CACHE Ø8 H, CACHE ØØ H)
CMI	(CM 24 H, CM 16 H, CM Ø8 H, CM ØØ H)
XB DECODE	(XBUF Ø8 H, XBUF ØØ H)

NOTE: XBUF 16 H is bi-directional; XBUF Ø8 H is input only.

2.1 Besides these external buses, two internal wired-or buses, the DBUS and the CBUS will be referred to in this description. At a given time, the DBUS may source one of the following:

A. WBUS
B. CMI DATA
C. CACHE DATA
D. Byte 1 of the XB DECODE Bus on byte Ø

Zeros on bytes 1, 2, and 3

The CBUS provides the input to the CMI drivers and may contain either an address from the CMI ADDRESS REGISTER or data from the WDR.

The clock available in the MDR chip is B CLK L. Registers and control signals ordinarily change on the low to high transitions of B CLK L.

2.2 **Chip I/O Summary:** Shall be as specified herein. See Para. 2.3 for Functional Descriptions.

CHIP I/O SUMMARY

Power/Ground	4 pins
WBUS (WB 24 H, WB 16 H, WB 08 h, WB 00 H)	4 pins
MBUS (-MB 24 H, -MB 16 H, -MB 08 H, -MB 00 H)	4 pins
MA BUS (MAD 24 H, MAD 16 H, MAD 08 H, MAD 00 H)	4 pins
PA BUS (PAD 16 H, PAD 08 H, PAD 00 H)	3 pins
CACHE BUS (CACHE 24 H, CACHE 16 H, CACHE 08 H, CACHE 00 H)	4 pins
CMI (CM 24 H, CM 16 H, CM 08 H, CM 00 H)	4 pins
XB DECODE BUS (XBUF 38 H, XBUF 00 H)	2 pins
B CLK L	1 pin
CHIP ID H	1 pin
MBUS ENA H	1 pin
ENA CMI L	1 pin
SNAPSHOT CMI L	1 pin
ADD REG ENA L	1 pin
XB SELECT H	1 pin
DBUS SEL S<1:0> H	2 pins
MMUX SEL S<1:0> H	2 pins
AMUX SEL S<1:0> H	2 pins
DBUS ROT S<1:0> H	2 pins
CLK SEL S<1:0> H	2 pins
PC <01:00> H	2 pins

2.3 **FUNCTIONS:**

WBUS: The WBUS may be sourced onto the DBUS from where it may be written to the WDR or the MDR, or driven to the PA BUS.

MBUS: The MBUS may source MA BUS, PA BUS, MDR or XB depending on the steering of the MBUS Multiplexer. The MBUS is active low.

MA BUS: The MA BUS may be sourced onto the MBUS or the PA BUS.

PA BUS: The PA BUS is a bi-directional bus and may be sourced or driven from the MDR chip. If driven externally, it may be sourced onto the MBUS or written to the CMI ADDRESS REGISTER. If driven from the MDR chip, it may contain the contents of the MA BUS, the DBUS, the CMI ADDRESS REGISTER, or the CMI DATA LATCH, depending on the steering of the PA BUS MULTIPLEXER.

CACHE BUS: The CACHE BUS is a bi-directional bus used to write data from the WDR to the cache or read data from the cache onto the DBUS.

CMI: The CMI is a bi-directional bus used to transfer data from the CBUS (either address from the CMI ADDRESS REGISTER or data from the WDR) to memory or data from memory to the DBUS or into the CMI DATA LATCH from where it may be sourced onto the PA BUS.

XB DECODE BUS: Byte 0 of the XB DECODE BUS is driven from byte 0 of the XB ROTATOR. Byte 1 of the XB DECODE BUS is bi-directional. It may be driven from byte 1 of the XB ROTATOR or it may source external data onto byte 0 of the DBUS.

B CLK L: Most state elements in the MDR chip change on the low to high transition of B CLK L. Certain control signals and the CMI data lines are latched during B CLK L low time to ensure that they do not change inside the MDR chip before the low to high transition of B
CLK L.

CHIP ID H: When CHIP ID H is high, the drivers for both bytes 0 and 1 of the PA BUS are continuously enabled (except when the PA BUS is being sourced onto the MBUS). When CHIP ID H is low, only the driver for byte 0 of the PA BUS is continuously enabled (except when the PA BUS is being sourced onto the MBUS).

MBUS ENA H: HIGH - MBUS drivers in the MDR chip are enabled
LOW - MBUS drivers in the MDR chip are disabled

ENA CMI L: The CMI drivers can only be enabled or disabled at the low to high transition of B CLK L. If ENA CMI L is LOW at the low to high transition of B CLK L, the CMI drivers will be enabled until the subsequent low to high transition of B CLK L. If ENA CMI L is HIGH at the low to high transition of B CLK L, the CMI drivers will be disabled until the subsequent low to high transition of B CLK L. In addition, the CMI driver for byte 3 can only be enabled when WDR is selected on the CBUS.

SNAPSHOT CMI L: While SNAPSHOT CMI L is high, the CMI DATA LATCH is transparent and passes data directly from the CMI to the PA BUS MULTIPLEXER and to the DBUS if CMI data is selected by the DBUS select controls. When SNAPSHOT CMI L is low at a low to high transition of B CLK L, the data that was on the CMI at the previous high to low transition of B CLK L is retained in the latch until the next low to high transition of B CLK L occurs with CMI DATA LATCH ENABLE high, at which time the latch becomes transparent again.

ADD REG ENA L: LOW - Data on the PA BUS is clocked into the CMI ADDRESS REGISTER at the HIGH to LOW transition of B CLK L.
HIGH - The CMI ADDRESS REGISTER is not changed.

XB SELECT H: LOW - XB0 <- (DBUS) at low to high transition of B CLK L if XB clocks are enabled.
HIGH - XB1 <- (DBUS) at low to high transition of B CLK L if XB clocks are enabled.
Also used in conjunction with PC <01:00> H to determine which bytes of each XB are to be read.
DBUS SEL S<1:0> H: DBUS SEL S<1:0> H control what data will be enabled onto the DBUS according to the following Table 2.

MDR Table 2

DBUS SEL S1 H	DBUS SEL S0 H	DBUS
LOW	LOW	CACHE DATA
LOW	HIGH	CMI DATA
HIGH	LOW	WBUS
HIGH	HIGH	XB DEC. BUS

When XB DECODE BUS is selected on the DBUS, XBUF 16 H is driven onto byte 0 of the DBUS, and the other three bytes are zeroed.

MMUX SEL S<1:0> H: MMUX SEL S<1:0> H determine what data will be sourced onto the MBUS from the MDR chip (if the MBUS drivers are enabled) according to the following Table 3.

MDR Table 3

MMUX SEL S1 H	MMUX SEL S0 H	MBUS
LOW	LOW	MDR
LOW	HIGH	XB DATA
HIGH	LOW	MA BUS
HIGH	HIGH	PA BUS

When the MBUS MULTIPLEXER is steered to PA BUS, the most significant bit will be a logical 1 (active low).

AMUX SEL S<1:0> H: AMUX SEL S<1:0> H determine what data will be sourced onto the PA BUS from the MDR chip (if the PA BUS drivers are enabled) according to the following Table 4:

MDR Table 4

AMUX SEL S1 H	AMUX SEL S0 H	PA BUS
LOW	LOW	CMI ADD REG
LOW	HIGH	CMI DATA
HIGH	LOW	MA BUS
HIGH	HIGH	DBUS

If PA BUS DRIVER ENABLE is false, the PA BUS MULTIPLEXER is steered to MA BUS for the least significant bit or bits which are continuously enabled. (See PA BUS DRIVER ENABLE under IMPLICIT CONTROL FUNCTIONS).

DBUS ROT S<1:0> H: DBUS ROT S<1:0> H cause data from the DBUS to appear on the inputs to the MDR and WDR byte rotated as shown in the following Table 5.

MDR Table 5

DBUS ROT S1 H	DBUS ROT S0 H	ROT OUT (BYTES)			
LOW	LOW	3	2	1	0
LOW	HIGH	0	3	2	1
HIGH	LOW	1	0	3	2
HIGH	HIGH	2	1	0	3

CLK SEL S<1:0> H: CLK SEL S<1:0> H determine which DBUS destinations will be clocked on the low to high transition of B CLK L according to the following Table 6.

MDR Table 6

CLK SEL S1 H	CLK SEL S0 H	ENABLE
LOW	LOW	NOTHING
LOW	HIGH	MDR
HIGH	LOW	XB
HIGH	HIGH	WDR

In addition to the conditions listed in the chart, portions of the MDR must be enabled for data returning from a READ, SECONC REFERENCE. A READ, SECOND REFERENCE is decoded in the MDR chip when:

```
        DBUS ROT S<1:0> H not equal to zero
and     CLK SEL S<1:0> H equal zero
and     WDR not being sourced onto the MBUS
```

Then, clocks are enabled for bytes of the MDR as shown in the following Table 7.

MDR Table 7

DBUS ROT S1 H	DBUS ROT S0 H	BYTES ENABLEC			
LOW	HIGH	3	x	x	x
HIGH	LOW	3	2	x	x
HIGH	HIGH	3	2	1	x

Any time the MDR or any portion of the MDR is enabled or XB is enabled, and CMI DATA is selected on the DBUS, the WDR is also enabled. In this case, the WDR MUX is steered to DBUS data instead of the DBUS ROTATOR output.

PC <01:00> H: PC <01:00> H and XB SELECT H determine which bytes of each XB will appear on the XB DECODE BUS, and on the MBUS if the MBUS MULTIPLEXER is steered to XB DATA, according to the following Table 8.

MDR Table 8

					XB DEC BYTE 1	XB DEC BYTE 0
XB. SEL H	PC 01 H	PC 00 H	MBUS BYTE 3	MBUS BYTE 2	MBUS BYTE 1	MBUS BYTE 0
LOW	LOW	LOW	XB1 BYTE 3	XB1 BYTE 2	XB1 BYTE 1	XB1 BYTE 0
LOW	LOW	HIGH	XB0 BYTE 0	XB1 BYTE 3	XB1 BYTE 2	XB1 BYTE 1
LOW	HIGH	LOW	XB0 BYTE 1	XB0 BYTE 0	XB1 BYTE 3	XB1 BYTE 2
LOW	HIGH	HIGH	XB0 BYTE 2	XB0 BYTE 1	XB0 BYTE 0	XB1 BYTE 3
HIGH	LOW	LOW	XB0 BYTE 3	XB0 BYTE 2	XB0 BYTE 1	XB0 BYTE 0
HIGH	LOW	HIGH	XB1 BYTE 0	XB0 BYTE 3	XB0 BYTE 2	XB0 BYTE 1
HIGH	HIGH	LOW	XB1 BYTE 1	XB1 BYTE 0	XB0 BYTE 3	XB0 BYTE 2
HIGH	HIGH	HIGH	XB1 BYTE 2	XB1 BYTE 1	XB1 BYTE 0	XB0 BYTE 3

2.4 Implicit Control Functions:

ZERO MDR:

If: XB DECODE BUS is selected on the DBUS
and DBUS ROT S<1:0> H not equal zero
and CLK SEL S<1:0> H equal zero

then zero the MDR on the low to high transition of B CLK L instead of writing byte 1 of the XB DECODE BUS into byte 0 of the MDR. In this case, CMI DATA will be enabled on the DBUS.

SOURCE WDR ONTO MBUS If: XB DECODE BUS is selected on the DBUS and DBUS ROT S<1:0> H not equal zero and CLK SEL S<1:0> H equal zero

then source the WDR onto the MBUS. In this case, CMI DATA will be enabled on the DBUS.

CACHE BUS DRIVER ENABLE: The CACHE BUS drivers are enabled when anything except CACHE DATA is selected on the DBUS.

PA BUS DRIVER ENABLE: The PA BUS drivers for all three bits are disabled when the PA BUS is being sourced onto the MBUS. Otherwise, the driver for byte 0 (if CHIP ID H is low) or bytes 0 and 1 (if CHIP ID H is high) are continuously enabled. The remaining drivers are disabled when:

CMI ADDRESS REGISTER is selected on the PA BUS
and (CMI ADDRESS REGISTER clock is enabled OR CACHE DATA is selected on the DBUS)

WDR MUX SELECT: Any time the MDR or any portion of the MDR is enabled and CMI DATA is selected on the DBUS, the WDR MUX is steered to DBUS data (unrotated). Otherwise, the mux is steered to DBUS ROTATOR output.

CBUS SELECT: The CMI ADDRESS REGISTER is quiescently selected on the CBUS. The low to high transition of B CLK L, subsequent to the transition which causes the CMI drivers to be enabled, will cause the WDR to be selected on the CBUS. The WDR will remain selected until the low to high transition of B CLK L subsequent to the transition which causes the CMI drivers to be disabled.

XB DECODE BUS DRIVER ENABLE: The driver for XB DECODE BUS byte 0 is continuously enabled. The driver for byte 1 is enabled except when:

XB DECODE BUS is selected on the DBUS
and DBUS ROT S<1:0> H equal zero

```
                               ---------------
     CLK SEL S1 H --->!01               48!<--- MBUS ENA H
     CLK SEL S0 H --->!02               47!---> MB 24 L
          B CLK L --->!03               46!---> MB 16 L
   ADD REG ENA L --->!04               45!<--- MAD 24 H
        ENA CMI L --->!05               44!---> MB 08 L
          PC 00 H --->!06               43!<--- MMUX SEL S1 H
          PC 01 H --->!07               42!---> MB 00 L
      XB SELECT H --->!08               41!<--- MMUX SEL S0 H
        XBUF 00 H <---!09               40!<--- MAD 08 H
 SNAPSHOT CMI L --->!10 ....... 39!<--- MAD 16 H
        XBUF 08 H <-->!11 .       . 38!----GROUND
              VGA ----!12 . LID . 37!<--- MAD 00 H
              VCC ----!13 . DOWN. 36!<--- CHIP ID H
   DBUS ROT S0 H --->!14 .       . 35!----GROUND
   DBUS ROT S1 H --->!15 ....... 34!<--- DBUS SEL S1 H
          CM 24 H <-->!16               33!<--- DBUS SEL S0 H
          CM 16 H <-->!17               32!<--> PAD 16 H
          CM 08 H <-->!18               31!<--> PAD 08 H
          CM 00 H <-->!19               30!<--> PAD 00 H
          WB 00 H --->!20               29!<--- AMUX SEL S1 H
          WB 08 H --->!21               28!<--- AMUX SEL S0 H
          WB 16 H --->!22               27!<--> CACHE 00 H
          WB 24 H --->!23               26!<--> CACHE 08 H
     CACHE 24 H <-->!24               25!<--> CACHE 16 H
                               ---------------
```

MDR FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.17 MASSBUS Register Control (MRC)

1.0 GENERAL DESCRIPTION

MRC is the MASSBUS register control chip. It produces two sets of encoded control signals. It contains bits of the control, status, and maintenance registers. It detects the initiation of data transfers, and produces the DT function codes. It produces the control signals for the MASSBUS control bus.

2. TERMINOLOGY

State elements will sometimes be described in terms of signals called "SET state name", "CLR state name", etc. If none of these signals is true, there is no change of state.

3. SIGNAL DESCRIPTIONS

3.1 MBA CLK L

MBA CLK L is a TTL input, and is the primary clock for MRC.

3.1.1 INIT L

INIT L is a TTL input. Some state elements are initialized by this signal, as described in later sections.

3.2 IE MAP <1:0> H, ADD 9-7 EQ L, ADD 6-5 EQ L, CMI FUN H, LATCHED ADD <4:2> H

These eight signals are TTL inputs. They are decoded to produce the various control outputs of MRC as described in later sections. They are decoded to produce internal signals as shown in the Table below.

IE MAP <1:0> H	ADD 9-7 EQ L	ADD 6-5 EQ L	LATCHED ADD <4:2>H	OUTPUT
0	0	0	0	SEL CONFIG H
0	0	0	1	SEL CTRL H
0	0	0	2	SEL STAT H
0	0	0	3	SEL VAR H
0	0	0	4	SEL BCR H
0	0	0	5	SEL MAINT H
0	0	0	7	SEL CMD/ADD H
1	X	X	X	SEL EXT H
2	X	X	X	SEL MAP H
0	0	0	X	INT REG H

3.3 MBA ADD L

MBA ADD L is a TTL input.

3.3.1 DO SLAVE OP

DO SLAVE OP is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

DO SLAVE OP <--- MBA ADD H * RETURN STAT L * END SLAVE L

where

END SLAVE H = DO SLAVE OP H * (SEL BCR H + SEL CONFIG H + SLAVE OP PGE H)

+ REG STATE H * (2 CYCLE RD H + SEL EXT H)

+ 2 CYCLE RD H * CMI FUN H + SEL CMD/ADD H * UPDATE CMD/ADD L

+ CP BUSY L * TRA SYNC L * CMI FUN H

where

2 CYCLE RD H = SEL STAT H + SEL CTRL H + SEL VAR H + SEL MAP H + SEL MAINT H

3.3.2 REG STATE

REG STATE is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

REG STATE <---(CHECK CTRL DATA H * SET PGE CTRL L + 2 CYCLE RD H * CMI FUN L) * DO SLAVE OP H + LD CBDB READ L + SET CB HUNG L

where

LD CBDB READ H = CP BUSY H * MDP TRA H * CTOD L.

3.3.3 SLAVE CONE H

SLAVE DONE H is a totem pole transciever.

SLAVE DONE H = END SLAVE H + RETURN STAT H

3.3.4 RETURN STAT H

RETURN STAT H is a totem pole output.

RETURN STAT H = DO SLAVE OP H * (INT REG H * (LATCHED ADD <4:2> H=6) + (IE MAP <1:0> 0) * ADD 9-5 EQ L)

3.4 MDIB SEL

MDIB SEL is an edge triggered flipflop, clocked on the rising edge of MBA CLK L.

LOAD MDIB = LD MAINT H

MDIb <--- IBUS 23 H

CLR MDIB = INIT H
A control state to both load and clear MBIB is illegal.

3.5 MM

MM is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD MM = LD CTRL H

MM <--- IBUS 3 H

CLR MM = INIT H
A control state to both load and clear MM is illegal.

3.6 IBC

IBC is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LOAD IBC = LD CTRL H

IBC <--- IBUS 4 H

CLR IBC = INIT H
A control state to both load and clear IBC is illegal.

3.7 MAP POINT SEL H

MAP POINT SEL H is a totem pole output.

MAP POINT SEL H = DT BUSY L + SEL MAP H * IBC H * DO SLAVE OP H * CMI FUN H

3.8 CTOD, CTOD H

CTOD is a transparent latch. It is open if LATCH EXT CS H * CP BUSY L.

CTOD <--- CMI FUN H * SET PGE EXT L

CTOD H is a totem pole output.

3.9 LATCH EXT CS

LATCH EXT CS is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

LATCH EXT CS <--- DO SLAVE OP H * SEL EXT H * CP IDLE H

3.9.1 CP BUSY, CP IDLE H

CP BUSY is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L. It is asynchronously cleared if INIT H * MBA CLK L.

SET CP BUSY = LATCH EXT CS H * SET PGE EXT L
CLR CP BUSY = (CLR TOUT CTR H + INIT H) * .NOT. SET CP BUSY

CP IDLE H = CP BUSY H * TRA SYNC L

3.9.2 TOUT CTR <3:0>

TOUT CTR <3:0> is a four bit synchronous up counter, clocked on the rising edge of MBA CLK L.

CLR TOUT CTR H = CP BUSY H * MDP TRA H + CP BUSY L * DO SLAVE OP L * SEL EXT L + TOUT CTR EQ 13 H + CP BUSY L * TRA SYNC L

COUNT TOUT CTR = CLR TOUT CTR L

3.9.3 TRA SYNC L

TRA SYNC L is a TTL input.

3.9.4 CB HUNG

CB HUNG is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET CB HUNG = CP BUSY L * TOUT CTR EQ 13 H

CLR CB HUNG =(LD STAT H * IBUS 23 H + INIT H + DT INIT H) * .NOT. SET CB HUNG

3.9.5 DEM, XMIT DEM H

DEM is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET DEM = CP BUSY H * (TOUT CTR <3:0> =1)

CLR DEM =(INIT H + (TOUT CTR <3:0> = 13) + (TRA H * RS EQ 4L)

* .NOT. SET DEM

XMIT DEM H is a totem pole output.

XMIT DEM H = DEM H

3.9.6 RS EQ 4

RS EQ 4 is a transparent latch, open if CP IDLE H * MBA CLK H.

RS EQ 4 <--- ADD 6-5 EQ H * (LATCHED ADD <4:2> H = 4)

3.9.7 NED

NED is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET NED = RS EQ 4 L * TOUT CTR EQ 13 h * CP BUSY H

CLR NED = (INIT L + LD STAT H * IBUS 18 H) * .NOT. SET NED

3.9.8 MDP TRA SYNC L

MDP TRA SYNC L is a totem pole output.

MDP TRA SYNC H = MDP TRA H * MBA CLK H

where,

MDP TRA H = (TOUT CTR <3:0> = 13) + LATCH EXT CS H * SLAVE OP PGE H + RS EQ 4 L * CP BUSY H * TRA SYNC H

3.10 UPPER VAR CNT EN L

UPPER VAR CNT EN L is a TTL input.

3.1Ø.1 UPDATE MAP

UPDATE MAP is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET UPDATE MAP = DT INIT H + UPPER VAR CNT EN H

CLR UPDATE MAP = (INIT H + CHECK MAP H) * .NOT. SET UPDATE MAP

3.1Ø.2 UPDATE CMD ACC

UPDATE CMD ADD is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SET UPDATE CMD ADD = DO CMI CYC L * DOING CMI CYC H + DT INIT H

CLR UPDATE CMD ADD = (INIT H + LD CMD ADD H) * .NOT. SET UPDATE CMD ADD

3.1Ø.3 DO CMI CYC H

DO CMI CYC H is an open collector transceiver. MRC drives DO CMI CYC H low if UPDATE CMD ADD H + UPDATE MAP H.

3.1Ø.4 DOING CMI CYC

DOING CMI CYC is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

DOING CMI CYC <--- DO CMI CYC H

3.11 DRIVE DT BUSY

DRIVE DT BUSY is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

DRIVE DT BUSY <--- LD EXT RØ H * IBUS 5 H * IBUS Ø H + INIT L * DT BUSY L

where,

LD EXT RØ H = LATCH EXT CS H * CMI FUN H * ADD 6-5 EQ H * (LATCHED ADD <4:2> = Ø)

3.11.1 DT BUSY H

DT BUSY H is an open collector transceiver. MRC drives DT BUSY H low if DRIVE DT BUSY L.

3.12 SECOND RD CYCLE

SECOND RD CYCLE is a transparent D latch, open when MBA CLK L is low.

SECOND READ CYCLE <--- DO SLAVE OP H * 2 CYCLE RD H * CMI FUN L + LD CBDB READ H + SET CB HUNG H

3.12.1 FORCE IBUS RW LO L

FORCE IBUS RW LO L is a totem pole output.

FORCE IBUS RW LO L = MBA CLK L + SECOND RD CYCLE L.

3.13 PGE

PGE is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

CLR PGE = INIT H + DT INIT H + LD STAT H * IBUS 15 H

SET PGE H = SET PGE EXT H + SET PGE MAINT H + SET PGE CTRL H + SET PGE VAR H + SET PGE BCT H + SET PGE MAP H

where,

SET PGE EXT H = IBUS 5 H * IBUS Ø H * DT BUSY H * LD EXT RØ H
SET PGE MAINT H = MM L * SEL MAINT H * DO SLAVE OP H * CMI FUN H SET PGE CTRL H = (SEL CTRL H * REG STATE L * CMI FUN H * DT BUSY H * DO SLAVE OP H) * (IB CL * IBUS 4 H + MM L * IBUS 3 H)
SET PGE VAR L = CMI FUN H * DT BUSY H * SEL VAR H * DO SLAVE OP H
SET PGE BCT L = CMI FUN H * DT BUSY H * SEL BCT H * DO SLAVE OP H
SET PGE MAP L = CMI FUN H * DT BUSY H * SEL MAP H * IBC L * DO SLAVE OP H

3.13.1 SLAVE OP PGE

SLAVE OP PGE is an edge triggered D flipflop, clocked on the rising edge of MBA CLK L.

SLAVE OP PGE <--- SET PGE H * SLAVE DONE L

3.14 IBUS n H

IBUS n H (n 23, 19, 18, 5:0) are nine tri-state transceivers. IBUS 31 H is a tri-state driver. Operation is summarized in the table below.

n	ENABLE	DATA
31	RD STAT H + INIT H	DT BUSY H + INIT H
23	RD MAINT H + RD STAT H + INIT H	CB HUNG H * RD STAT H + MDIB SEL H * RD STAT L
19	RD STAT H + INIT H	RD STAT H * PGE H
18	RD STAT H + INIT H	RD STAT H * NED H
5	INIT H	1
4	RD CTRL H + INIT H	IBC H * INIT L
3	INIT H	0
2	INIT H	1
1	INIT H	0
0	INIT H	0

3.15 DT FUN <1:0>, DT DIR,

DT FUN <1:0> and DT DIR are three edge triggered master slave flipflops. The input stage is clocked on the rising edge of MBA CLK L. The outputs change with the falling edge of MBA CLK L.

```
LOAD DT FUN 1 = LD EXT R0 H
     DT FUN 1 <--- IBUS 4 H + IBUS 3 H
     CLR DT FUN 1 = INIT H + DRIVE DT BUSY H * DT BUSY L

LOAD DT FUN 0 = LD EXT R0 H
     DT FUN 0 <--- IBUS 3 H
     CLR DT FUN 0 = INIT H + DRIVE DT BUSY H * DT BUSY L

LOAD DT DIR = LD EXT R0 H
     DT DIR <--- IBUS 2 H * IBUS 1 H
     CLR DT DIR = INIT H
```

3.15.1 DT IDLE

DT IDLE is a transparent latch, open when MBA CLK L is high. DT IDLE H <--- (DT FUN <1:0> H = 0)

3.15.2 DT INIT H

DT INIT H = DT IDLE H * (DT FUN <1:0> H O)

3.16 REG CTRL GA <2:0> H, IBUS RW H

REG CTRL GA <2:0> H AND IBUS RW H are totem pole transceivers.

REG CTRL GA 2 H = DO SLAVE OP H * SET PGE BCL L * SET PGE MAINT L * (SEL CTRL H + SEL MAINT H + SEL CTRL H) + CHECK MAP H + INIT H

REG CTRL GA O L = (DO SLAVE OP H * SET PGE VAR L * SET PGE MAINT L * SET PGE MAP L * (SEL VAR H + SEL MAINT H + SEL MAP H) + UPDATE CMD ADD H * UPDATE EN H + DO SLAVE OP H * SEL EXT H * CP IDLE H) * INIT H

IBUS RW H = DO SLAVE OP H * CMI FUN L * (SEL CTRL H + SEL STAT H + SEL VAR H + SEL MAINT H + SEL MAP H) + CP BUSY H * TRA SYNC H * CTOD L * RS EQ 4 L + UPDATE CMD ADD * UPDATE EN H + REG STATE H * DO SLAVE OP H * CMI FUN L * SEL ET H + INIT H

where,

CHECK MAP H = UPDATE MAP H * UPDATE EN H + DO SLAVE OP H * SEL MAP H * SET PGE MAP L.

and

UPDATE EN H = DO SLAVE OP L + SEL CMD ADD H * CMI FUN L

3.16.1 INTERNAL USE OF REG CTRL GA <2:0> H

MRC uses the received versions of REG CTRL GA <1:0> H and IBUS RW H to load and read internal state elements. The following table defines the control signals.

IBUS RW H	REG CTRL GA <2:0> H	SIGNAL
0	3	LD STAT H
1	3	RD STAT H
0	4	LD MAINT H
1	4	RD MAINT H
0	5	LD CTRL H
1	5	RD CTRL H
1	0	LD CMD ADD H
1	6	CHECK MAP H

3.17 REG CTRL MDP <2:0> H

REG CTRL MDP <2:0> H are totem pole outputs.

REG CTRL MDP 2 H = DO SLAVE OP H * [REG STATE H * CMI FUN L * SEL EXT H + CMI FUN H * CP IDLE H * SEL EXT H + (SEL STAT H + SEL MAINT A) * CMI FUN L + SEL BCR H * SET PGE BCT L + UPDATE CMD ADD L * SEL CMD ADD H]

REG CTRL MDP 1 H = DO SLAVE OP H * [(REG STATE H * CMI FUN L + CMI FUN H * CP IDLE H) * SEL EXT H + SEL CTRL H + SEL MAINT H * CMI FUN L * MDIB SEL H * SET PGE MAINT L + CMI FUN H * SET PGE VAR L * SET PGE MAINT L * (SEL VAR H + SEL MAINT H) + SEL CMD ADD H * UPDATE CMD ADD L + CMI FUN H * SEL STAT H] + UPDATE EN H * UPDATE CMD ADD H * UPDATE MAP L * LD CBDB READ L.

REG CTRL MDP 0 H = DO SLAVE OP H * [(REG STATE H * CMI FUN L + CMI FUN H * CP IDLE H) * SEL EXT H + SEL STAT H + SEL CTRL H * DT BUSY H * CMI FUN H * REG STATE L + CMI FUN H * (SEL CTRL H + SET PGE MAINT L * SET PGE VAR L * (SEL MAINT H + SEL VAR H) + SEL BCR H * SET PGE BCT L) + SET PGE MAP L * SEL MAP H] + UPDATE EN H * LD CBDB READ L * UPDATE CMD ADD H.

3.18 SILO CYCLE STALL L

SILO CYCLE STALL L is a totem pole output.

SILO CYCLE STALL H = CHECK MAP H + MDP TRA H * CTOD L * CP BUSY H + DO SLAVE OP H * CP IDLE H * RW EXT H.

MRC Table 1

Pin Identification and Gate Type

PIN #	PIN ID	GATE TYPE
1	IBUS 4 H	GA1TZF
2	FORCE IBUS RW LO L	GA1TTN
3	IBUS 3 H	GA1TZF
4	SLAVE DONE H	GA1TTF
5	DO CMI CYC H	GA1TPF
6	RETURN STAT H	GA1TTN
7	DT FUN 1 H	GA1TTN
8	INIT L	GA1TNF
9	UPPER VAR CNT EN L	GA1TNF
10	MBA ADD L	GA1TNF
11	MAP POINT SEL H	GA1TTN
14	CMI FUN H	GA1TNF
15	LATCHED ADD 4 H	GA1TNF
16	IE MAP 1 H	GA1TNF
17	ADD 9-7 EQ L	GA1TNF
18	LATCHED ADD 3 H	GA1TNF
19	IE MAP 0 H	GA1TNF
20	LATCHED ADD 2 H	GA1TNF
21	ADD 6-5 EQ L	GA1TNF
22	REG CTRL MDP 1 H	GA1TTN
23	REG CTRL MDP 2 H	GA1TTN
24	REG CTRL MDP 0 H	GA1TTN
25	TRA SYNC L	GA1TNF
26	CTOD H	GA1TTN
27	MDP TRA SYNC L	GA1TTN
28	MBA CLK L	GA1TNF
29	DT DIR H	GA1TTN
30	SILO CYCLE STALL L	GA1TTN
31	IBUS 5 H	GA1TZF
32	IBUS 2 H	GA1TZF
33	IBUS 1 H	GA1TZF
34	XMIT DEM H	GA1TTN
36	IBUS 18 H	GA1TZF
37	REG CTRL GA 2 H	GA1TTF
39	REG CTRL GA 0 H	GA1TTF
40	IBUS 23 H	GA1TZF
41	IBUS RW H	GA1TTF
42	DT BUSY H	GA1TPF
43	IBUS 19 H	GA1TZF
44	IBUS 31 H	GA1TZN
45	REG CTRL GA 1 H	GA1TTF
46	INTER L	GA1TPN
47	DT FUN 0 H	GA1TTN
48	IBUS 0 H	GA1TZF

MRC FIGURE 1

PIN CONFIGURATION DIAGRAM

```
                                   ---------------

            IBUS 4 H <-->!01               48!<--> IBUS 0 H
  FORCE IBUS RW LO L <---!02               47!---> DT FUN 0 H
            IBUS 3 H <-->!03               46!---> INTER L
        SLAVE DONE H <-->!04               45!<--> REG CTRL GA 1 H
        DO CMI CYC H <-->!05               44!---> IBUS 31 H
       RETURN STAT H <---!06               43!<--> IBUS 19 H
          DT FUN 1 H <---!07               42!<--> DT BUSY H
              INIT L --->!08               41!<--> IBUS RW H
   UPPER VAR CNT EN L --->!09              40!<--> IBUS 23 H
           MBA ADD L --->!10 .......       39!<--> REG CTRL GA 0 H
    MAP POINT SEL H  <---!11 .       .     38!---- GROUND
                 VGA ----!12 . LID .       37!<--> REG CTRL GA 2 H
                 VCC ----!13 . DOWN.       36!<--> IBUS 18 H
           CMI FUN H --->!14 .       .     35!---- GROUND
     LATCHED ADD 4 H  --->!15 .......      34!---> XMIT DEM H
          IE MAP 1 H --->!16               33!<--> IBUS 1 H
       ADD 9-7 EQ L --->!17                32!<--> IBUS 2 H
     LATCHED ADD 3 H  --->!18              31!<--> IBUS 5 H
          IE MAP 0 H --->!19               30!---> SILO CYCLE STALL L
     LATCHED ADD 2 H  --->!20              29!---> DT DIR H
       ADD 6-5 EQ L --->!21                28!<--- MBA CLK L
     REG CTRL MDP 1 H <---!22              27!---> MDP TRA SYNC L
     REG CTRL MDP 2 H <---!23              26!---> CTOD H
     REG CTRL MDP 0 H <---!24              25!<--- TRA SYNC L

                                   ---------------
```

3.18 MASSBUS SILO CONTROL (MSC)

1.0 General

MSC is the MASSBUS ADAPTER SILO CONTROL CHIP. It contains 2 counters for addressing 32 bytes of SILO RAM, and logic for detecting SILO empty and SILO full. It controls the generation and checking of SILO and MASSBUS DATA BUS parity. Bits <1:0> of the VAR are resident in MSC and are used for BYTE CONTROL. The CMI DATA MASK is generated in MSC. . MSC generates the control field for controlling the flow of data in the MDP's (The data path chips.)

2. Terminology

Some state elements are described in terms of signal names SET statename or CLR statename. If SET statename is true, the element is set; if CLR statename is true, the element is cleared. If none are true, there is no change of state.

A signal name written as "Signal Name IN (H or L)" is the output of the input stage of a D flipflop clocked on the rising edge MBA CLK L. This signal is the equivalent of a D latch, open when MBA CLK L is low, and contains the state to which the flipflop will be set on the rising edge of the clock.

3. INTERNAL REGISTERS - IBUS n H

MSC contains the following bits of MBA INTERNAL REGISTERS. These are loaded from and read to the IBUS.

REGISTER	BIT	IBUS BIT
VAR	VAR <1:0>	<1:0>
STATUS	MDPE	6
STATUS	SPE	14
STATUS	DLT	11
MAINT.	INV DPG	31
MAINT.	INV SPG	22

IBUS n H is a tri-state port. Bits are described fully in later sections.

4. DETAILED CHIP DESCRIPTION

4.1 SILO CONTROL

There are two counters used for loading and reading the SILO RAM - SMCTR (SILO- MASSBUS COUNTER) and SCCTR (SILO-CMI COUNTER). Two flipflops are used to help detect SILO full and SILO empty. These are 2 BYTE EN and JDSC ("Just done Silo CMI XFER"). JDSM (Just done Silo Massbus XFER") EQUALS .NOT. JDSC.

4.1.1 SC CTR <4:Ø>

SC CTR is a five bit synchronous counter. It changes state with the rising edge of MBA CLK L.

SC CTR is incremented if SC XFER H

SC CTR is cleared if DT INIT H + INIT H

4.1.2 SM CTR <4:1>

SM CTR is a four bit synchrounous counter. It changes state with the rising edge of MBA CLK L.

SM CTR is incremented if SM XFER H.

SM CTR is cleared if DT INIT H + INIT H

4.1.3 CTRS <4:1> EQ H

CTRS <4:1> EQ IN H = [SC CTR <4:1> IN H = SM CTR <4:1> IN H].

CTRS <4:1> EQ H is the output of a D latch, Open when 2ND MBA CLK L is high.

CTRS <4:1> EQ H <- CTRS <4:1> EQ IN H.

4.1.4 2 BTYE EN

2 BYTE EN is an edge flip flop clocked on the rising edge of MBA CLK L.

SET 2 BYTE EN = (SC XFER H + INIT FOR RD H) * .NOT. CLR 2 BYTE EN.

CLR 2 BYTE EN = INIT for W/WCK H + RESET 2 BYTE EN H.

4.1.4.1 RESET 2 BYTE EN

RESET 2 BYTE EN is a D latch, open when 2ND MBA CLK L is high.

RESET 2 BYTE EN <- JDSM IN H * CTRS 5:1 IN EQ H * INH SM+SC L *(DO CMI CYC IN H + (SC CTR Ø IN L * SC XFER IN INH L))

where

INH SM+SC H = DT IDLE H + DT FUN Ø H * OLD SCLK IN H

SC XFER IN INH H = DO CMI CYC IN H + (DO FILL IN L * SC DONE IN H)

4.1.5 JDSC - JDSM

JDSC is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
JDSM = .NOT. JDSC

SET JDSC = [SC XFER H + INIT FOR RD H] * .NOT. CLR JDSC
CLR JDSC = SM XFER H + INIT FOR W/WCK H
```

4.1.6 SC XFER

SC XFER is a D latch, open when 2ND MBA CLK L is high.

```
SC XFER <- SC XFER IN INH L * SM XFER L * INH SM+SC L *
(CTRS 5:1 IN EQ L + JDSM IN H + SC CTR 0 IN H).
```

4.1.7 SM XFER

SM XFER is a D LATCH, open when 2ND MBA CLK L is high.

```
SM XFER <- MDPE L * INH SM+SC L * SMDB READY IN  L * 2
BYTE EN IN H * (CTRS 5:1 IN EQ L + JDSM IN L).
```

4.1.8 SILO ADDRESS <4:1> H, SILO CHIP SEL <1:0> L

SILO ADDRESS <4:1> H and SILO CHIP SEL <1:0> are totem pole OUTPUTS.

SILO ADDRESS <4:1> H is the output of a multiplexer.

```
if SM XFER H, SILO ADD <4:1> H <- SM CTR <4:1>
if SC XFER H, SILO ADD <4:1> H <- SC CTR <4:1>
Else,         SILO ADD <4:1> H <- 0

SILO CHIP SEL 0 L is asserted low if:
     SM XFER H + (SC XFER H * VAR 0 L)

SILO CHIP SEL 0 L is asserted low if:
     SM XFER H + (SC XFER H * VAR 0 H)
```

4.2 DT FUN <1:0> H - DT IDLE

DT FUN <1:0> H are used to define the data transfer function. They are encoded as follows:

00	IDLE
01	Read from drive (DT READ)
10	Write to drive (DT WRITE)
11	Write check (DT WCK)

These inputs must not change except when MBA CLK L is low.

DT IDLE is a transparent latch, open when MBA CLK L is high. It latches the value DT FUN 1 L * DT FUN 0 L.

DT IDLE is anded with DT FUN <1:0> as shown in table 3 to generate initialize signals.

MSC TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	GATE TYPE
1	CMI 30 H	GA1TTN
2	CMI 31 H	GA1TTN
3	CMI 28 H	GA1TTN
4	CMI 29 H	GA1TTN
5	IBUS 11 H	GA1TZF
6	IBUS 6 H	GA1TZF
7	REG CTRL GA 0 H	GA1TNF
8	DO CMI CYC H	GA1TCF
9	SCLK SYNC L	GA1TNF
10	SILO CTRL MDP 1 H	GA1TTN
11	SILO CTRL MDP 0 H	GA1TTN
14	START STOP MB L	GA1TTF
15	REG CTRL GA 2 H	GA1TNF
16	IBUS RW H	GA1TNF
17	DT DIR H	GA1TNF
18	IBUS 0 H	GA1TZF
19	2ND MBA CLK L	GA1TNF
20	IBUS 1 H	GA1TZF
21	SILO CTRL MDP 2 H	GA1TTN
22	SILO CHIP SEL 1 L	GA1TTN
23	MBA CLK DLY H	GA1TNF
24	DT FUN 0 H	GA1TNF
25	EXC SYNC L	GA1TNF
26	REG CTRL GA 1 H	GA1TNF
27	LAST TRANS H	GA1TNF
28	SILO ADDRESS 3 H	GA1TTN
29	SILO CHIP SEL 0 L	GA1TTN
30	MBA CLK L	GA1TNF
31	SILO ADDRESS 4 H	GA1TTN
32	SILO ADDRESS 2 H	GA1TTN
33	SILO ADDRESS 1 H	GA1TTN
34	DT FUN 1 H	GA1TNF
36	LOAD SILO H	GA1TTN
37	SILO PAR 0 H	GA1TZF
39	SILO PAR 1 H	GA1TZF
40	PAR L	GA1TNF
41	IBUS 22 H	GA1TZF
42	GATED SCLK H	GA1TNF
43	IBUS 31 H	GA1TZF
44	MM L	GA1TNF
45	MB DPA H	GA1TZF
46	IBUS 14 H	GA1TZF
47	ABORT L	GA1TCN
48	CMI BYTE CNTR EQ 1 L	GA1TNF

Table 2.

DT IDLE H	0	1	1	1	1
DT FUN 1 H	X	0	0	1	1
DT FUN 0 H	X	0	1	0	1
INIT FOR RD H	0	0	1	0	0
INIT FOR W/WCK H	0	0	0	1	1
DT INIT H	0	0	1	1	1
INIT FOR WRITE H	0	0	0	1	0

4.3 DT DIR H

DT DIR H defines data transfer direction

DT FWD H = DT DIR H
DT REV H = DT DIR L

DT DIR affects the VAR, MASK, and associated logic.

4.4 REG CTRL GA n H - IBUS RW H

REG CTRL GA <2:0> H and IBUS RW H are used together to read and write registers contained in MSC. Codes are explicitly described in following sections. Some codes are unused by the chip, but affect other MBA chips.

The internal signal, INIT H is defined by these control signals.

INIT H = (REG CTRL GA = 7) * IBUS RW H

4.5 VAR

MSC contains a two bit loadable up/down synchronous counter named VAR <1:0>. All transitions in state occur with the rising edge of MBA CLK L. VAR <1:0> is loaded from IBUS <1:0>.

LOAD VAR = (REG CTRL GA = 2) * IBUS RW L
COUNT UP VAR = SC XFER H * DT FWD H
COUNT DOWN VAR = SC XFER H * DT REV H
SC XFER H * LOAD VAR is an illegal condition.

VAR <1:0> is read to IBUS <1:0> H when
(REG CTRL GA = 2) * IBUS RW H

4.6 XMIT DO CMI CYC

XMIT DO CMI CYC is an edge triggered D flop clocked on the rising edge of MBA CLK L.

SET XMIT DO CMI CYC = INIT FOR W/WCK H + SPE IN L * SC XFER H [DT FWD H * (DT FUN <1:0> H = 3) + DT REV H * (DT RUN 1:0 H = 0)] * .NOT. (DT FUN 1 H * SC DONE IN H) + DT READ H * SC DONE IN H * SC DONE L.

CLR XMIT DO CMI CYC = (INIT FOR RD H + CLR MASK H) * .NOT. SET XMIT DO CMI CYC.

4.6.1 DO CMI CYC H

DO CMI CYC H is an open collector bi-directional signal. It is pulled low by MSC if XMIT DO CMI CYC L.

4.6.2 DOING CMI CYC

DOING CMI CYC is an edge triggered D flipflop clocked on the rising edge of MBA CLK L.

DOING CMI CYC <- DO CMI CYC H.

4.6.3 CLR MASK H

CLR MASK H = DT IDLE H + DO CMI CYC L * DOING CMI CYC H.

4.7 MASK <3:0>

MASK <3:0> are four transparent latches, which are open when MBA CLK L is low.

MASK n is latched set if SC XFER H * <VAR <1:0> = n)

It is latched clear if CLR MASK H + INIT FOR RD H

Otherwise, it is latched with it's previous value.

4.8 CMI MASK <31:28> H

CMI MASK <3:0> H are four totempole outputs.

CMI MASK n H = MASK n H + DT FUN 0 L.

4.9 SCLK SYNC L - LATCHED SCLK SYNC

LATCHED SCLK SYNC is an edge triggered flipflop clocked on the rising edge of MBA CLK L. LATCH SCLK SYNC <- SCLK SYNC H. It is asynchronously cleared if (DT INIT H + INIT H) * MBA CLK H.

4.10 SMDB READY

SMDB READY is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
SET SMDB READY =  SM XFER H * .NOT. (CLR SMDB READY)
CLR SMDB READY = LATCHED SCLK SYNC
```

It is asynchronously set if INIT FOR RD H * MBA CLK H.

It is asynchronosuly cleared if INIT FOR W/WCK H * MBA CLK H.

4.11 DLT

DLT is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
SET DLT = SCLK SYNC H * SMDB READY L * LATCHED LAST
TRANS H
    * .NOT. (DT INIT H + INIT H)
CLR DLT = (DT INIT H + INIT H) + [LD STATUS H
    * IBUS 11 H * .NOT. SET DLT]
```

DLT is read to IBUS 11 H if (REG CTRL GA=3) * IBUS RW H

4.12 LATCHED LAST TRANS - LAST TRANS H

LATCHED LAST TRANS is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
SET LATCHED LAST TRANS = LAST TRANS H
CLR LATCHED LAST TRANS = DT INIT H + INIT H
```

4.13 SC DONE

SC DONE is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

```
SET SC DONE = (SC XFER H * CMI BYTE CNT EQ -1 H + EXC
SYNC H
                      * CTRS <5:1> EQ H * JDSC H + SPE H)
                      * .NOT. CLR SC DONE

CLR SC DONE = DT IDLE H
```

4.14 DO FILL

DO FILL is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

CLR DO FILL = SC XFER H * SC DONE H * SC CTR 0 H

It is asynchronously set if INIT FOR W/WCK H * MBA CLK H.

It is asynchronously cleared if INIT FOR READ H * MBA CLK H.

4.15 START STOP MB

START STOP MB L is the inverted output of an asynchronously set and reset latch.

SET START STOP MB = 2ND MBA CLK H * MBA CLK DLY H * CTRS <5:1> EQ H * JDSC H * SC CTR 0 L * DT IDLE L + SC DONE) * DT FUN 1 H] + DT READ H * LAST TRANS H.

CLR START STOP MB = DT FUN 1 H * LAST TRANS H + SC DONE H * DO CMI CYC L * DT READ H + DT IDLE H.

If a SET and CLR condition occur simultaneously, START STOP MB L
will be high.

4.16 PARITY

4.16.1 MSC contains logic for control of MASSBUS DATA BUS and SILO PARITY. MSC contains the bits of the maintenance register that control inverting MASSBUS DATA BUS and SILO parity and for storing parity error information.

4.16.2 PAR H

PAR H is the one signal defining the odd PARITY of the data being handled by the MDP chips. MSC uses PAR H for storing and checking parity.

4.16.3 INV SPG, INV DPG

INV SPG and INV DPG are transparent latches. They are asynchronously cleared if INIT H. They are open if MBA CLK H * MBA CLK DLY H * (REG CTRL GA = 4) * IBUS RW L

INV SPG <- IBUS 22 H
INV DPG <- IBUS 31 H

INV SPG is read to IBUS 22 H and
INV DPG is read to IBUS 31 H if
(REG CTRL GA = 4) * IBUS RW H

4.16.4 SC PAR

SC PAR is an edge triggered flipflop clocked on the rising edge of MBA CLK L. If SC XFER H, it is loaded with PAR H ● SP G L.

Else, SC PAR is unchanged.

4.16.5 SM PAR

SM PAR is an edge triggered flipflop clocked on the rising edge of MBA CLK L,

If * DT FUN 1 H * SM XFER H,
then SM PAR <- PAR H

If * DT READ H * LATCHED SCLK SYNC H
then SM PAR <- DPA H

else SM PAR unchanged.

4.16.6 DPA

DPA is an edge triggered flipflop clocked with the falling edge of GATED SCLK H.

If DT FUN 0 H, DPA <- MB DPA H

iF DT WRITE H, DPA <- SM PAR H ● INV DPG H

else DPA unchanged.

4.16.7 MB DPA H

MB DPA H is a tristate signal. It is enabled if MM H + DT FUN 0 H. When enabled, MB DPA H <- DPA H.

4.16.8 SILO PARITY <1:0> H , LOAD SILO H

SILO PARITY <1:0> H are tristate signals. They are enabled by the signal that drives the totem pole output, LOAD SILO H.

LOAD SILO H = DT READ H * SM XFER H + DT FUN 1 H * SC XFER H.

When enabled

SILO PAR 0 H <- DT FUN 1 H * PAR H

SILO PAR 1 H <- DT READ H * SM PAR L + DT FUN 1 H * (PAR H * SPG L + PAR L * SC CTR 0 L * INV SPG H)

4.16.9 MDPE

MDPE is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

CLR MDPE = [(REG CTRL GA = 3) * IBUS RW L * IBUS 6 H +
INIT H +
DT INIT H] * .NOT. SET MDPE

SET MDPE = (PAR H ● DPA H) * LATCHED SCLK SYNC H * DT FUN 0 H

MDPE is read to IBUS 6 H if (REG CTRL GA = 3) * IBUS RW H

4.16.10 SPE

SPE is an edge triggered flipflop clocked on the rising edge of MBA CLK L.

CLR SPE = [(REG CTRL GA = 3) * IBUS RW L * IBUS 14 H + INIT H + DT INIT H] * .NOT. SET SPE

SET SPE = (PAR L ● SILO PAR 1 H ● SC PAR H) * SC XFEK H * SC CTR 0 H * DT READ H + (PAR L ● SILO PAR 1 H ● SILO PAR 0 H) * SM XFER H * (DT WRITE H + DT WCK H)

SPE is read to IBUS 14 if (REG CTRL GA = 3) * IBUS RW H

4.17 SILO CTRL MDP <2:0> H

SILO CTRL MDP <2:0> H are 3 totem pole outputs.

SILO CTRL MDP 2 H = SC XFER H * (DT READ H + DT FUN 1 H * SC DONE E L).

SILO CTRL MDP 1 H = DT FUN 0 H * LATCHED SCLK SYNC H + SC XFER H * (DT READ H + DT FUN 1 H * SC DONE L) * VAR 1 H + DT FUN 1 H * SM XFER H.

SILO CTRL MDP 0 = SC XFER H * (DT READ H + DT FUN 1 H * SC DONE L)* VAR 0 H + DT READ H * (LATCHED SCLK SYNCH + SM XFER H) + DT FUN 1 H * SC XFER H * SC DONE H * (DO FILL H + SC CTR 0 L)

4.18 ABORT L

ABORT L is an open collector signal. It is asserted low if MDPE H + SPE H + DLT H.

MSC TABLE 3 I/O PINS

SIGNAL NAME	#	DIRECTION	TYPE
DT FUN n H n = 1:0	2	I	
DT DIR H	1	I	
MBA CLK L	1	I	
IBUS n H n = 31,22,14,11, 6,1,0	7	I/O	TRISTATE
SILO ADDRESS n H n = 5:1	5	O	TOTEM POLE
SILO CHIP SEL n L n = 1:0	2	O	TOTEM POLE
LOAD SILO H	1	O	TOTEM POLE
SILO PAR n H n = 1:0	2	I/O	TRISTATE
PAR H	1	I	
ABORT L	1	O	O.C.
SILO CYCLE STALL L	1	I	
GATED SCLK H	1	I	
MM L	1	I	
MB DPA H	1	I/O	TRISTATE
SCLK SYNC L	1	I	
MB BYTE CNT EQ -1 H	1	I	
LATCHED EXC L	1	I	
START STOP MB L	1	O	TOTEM POLE
REG CTRL GA n H n = 2:0	3	I	
IBUS RW H	1	I	
SILO CTRL MCP n H n = 2:0	3	O	TOTEM POLE
DO CMI CYC H	1	I/O	O.C.
CMI n H n = 31:28	4	O	TRISTATE
CMI OBUF CMD EN L	1	I	
TOTAL	44		

MSC FIGURE 1

PIN CONFIGURATION DIAGRAM

```
                            --------------
          CMI 30 H <---!01         48!<--- CMI BYTE CNTR EQ 1 L
          CMI 31 H <---!02         47!---> ABORT L
          CMI 28 H <---!03         46!<--> IBUS 14 H
          CMI 29 H <---!04         45!<--> MB DPA H
         IBUS 11 H <-->!05         44!<--- MM L
          IBUS 6 H <-->!06         43!<--> IBUS 31 H
  REG CTRL GA 0 H  --->!07         42!<--- GATED SCLK H
     DO CMI CYC H <-->!08          41!<--> IBUS 22 H
      SCLK SYNC L --->!09          40!<--- PAR L
SILO CTRL MDP 1 H <---!10 ....... 39!<--> SILO PAR 1 H
SILO CTRL MDP 0 H <---!11 .     . 38!---- GROUND
              VGA ----!12 . LID . 37!<--> SILO PAR 0 H
              VCC ----!13 . DOWN. 36!---> LOAD SILO H
 START STOP MB L  <-->!14 .     . 35!---- GROUND
 REG CTRL GA 2 H  --->!15 ....... 34!<--- DT FUN 1 H
        IBUS RW H --->!16          33!---> SILO ADDRESS 1 H
         DT DIR H --->!17          32!---> SILO ADDRESS 2 H
         IBUS 0 H <-->!18          31!---> SILO ADDRESS 4 H
    2ND MBA CLK L.--->!19          30!<--- MBA CLK L
         IBUS 1 H <-->!20          29!---> SILO CHIP SEL 0 L
SILO CTRL MDP 2 H <---!21          28!---> SILO ADDRESS 3 H
SILO CHIP SEL 1 L <---!22          27!<--- LAST TRANS H
    MBA CLK DLY H --->!23          26!<--- REG CTRL GA 1 H
       DT FUN 0 H --->!24          25!<--- EXC SYNC L
                            --------------
```

3.19 MICRO SEQUENCER (MSQ) DC621

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for the VAX-11/750 micro-sequencer.

MSQ TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE	PIN #	PIN ID	COMMENTS	GATE TYPE
1	LIT 1 H		GA1TNF	24	NEXT 00 H		GA1TNF
2	LD OSR L		GA1TTN	25	CS ADDR 02 L	I_{OL}=12MA	GA1TCN
3	BUT CONTROL CODE H		GA1TNH	26	ENABLE UVECT H		GA1TNF
4	BUT 2 H		GA1TNF	27	CS ADDR 03 L	I_{OL}=12MA	GA1TCN
5	BUT 0 H		GA1TNF	28	NEXT 03 H		GA1TNF
6	BUT 1 H		GA1TNF	29	USTK D04 H		GA1TNF
7	LIT 0 H		GA1TNF	30	USTK D03 H		GA1TNF
8	FPA WAIT L		GA1TTN	31	CS ADDR 04 L	I_{OL}=12MA	GA1TCN
9	USTK ADDR 3 H		GA1TTN	32	NEXT 04 H		GA1TNF
10	USTK ADDR 1 H		GA1TTN	33	NEXT 05 H		GA1TNF
11	USTK ADDR 2 H		GA1TTN	34	CS ADDR 05 L	I_{OL}=12MA	GA1TCN
14	USTK ADDR 0 H		GA1TTN	36	JSR H		GA1TNF
15	IRD CTR 1 H		GA1TNF	37	USTK OUT ENABLE L		GA1TTN
16	IRD CTR 2 H		GA1TNF	39	ENABLE IRD ROM H		GA1TTN
17	USTK D00 H		GA1TNF	40	MICRO ADDR INH L		GA1TNF
18	USTK D01 H		GA1TNF	41	ZERO HI NEXT L		GA1TTN
19	NEXT 01 H		GA1TNF	42	DISABLE OSR BUT L		GA1TNF
20	CS ADDR 00 L	I_{OL}=12MA	GA1TCN	43	USTK D05 H		GA1TNF
21	CS ADDR 01 L	I_{OL}=12MA	GA1TCN	44	M CLK L		GA1TNF
22	USTK D02 H		GA1TNF	45	MSEQ INIT L		GA1TNF
23	NEXT 02 H		GA1TNF	46	DO SRVC L		GA1TNF
				47	B CLK L		GA1TNF
				48	DISABLE HI NEXT H		GA1TTN

2. Performance:

2.1 State Elements:

2.1.1 USP <3:0>:

This register is a 4 bit counter built from D type flipflops. It is clocked with the rising edge of M CLK L. The register is direct cleared when Internal MSEQ INIT L is asserted (see 3.3.2.6).

```
If JSR H, ENABLE UVECT H, or DO SRVC L is asserted then
    USP <3:0> = USP <3:0> + 1
```

If the conditions described in the previous paragraph are not true and BUT CONTROL CODE H is asserted, and

```
    (BUT<2:0> = 2, 3 or
    (BUT<2:0> = 1 and IRD CTR<2:1> ≠ 0)) then
    USP <3:0> = USP <3:0> - 1
```

2.2 Pin Descriptions:

Certain input pins are defined as having "Skew Latches". These are feed through latches that are open when the clock is not asserted and closed when the clock is asserted.

2.2.1 CS ADDR <05:00> L:

These are the address lines used to access the Control Store ROMs. They can be generated inside the chip or externally.

See Table I.

2.2.2 NEXT <05:00> H:

These signal lines are used to form CS ADDR<05:00> L.

2.2.3 USTK D<05:00> H:

These signal lines are used to form CS ADDR<05:00> L.

2.2.4 M CLK L:

This is the clock associated with the micro-sequencer. The rising edge of M CLK is used to clock the USP, and to specify:

LD OSR L
USTK OUT ENABLE L
DISABLE HI NEXT H
ZERO HI NEXT L

2.2.5 B CLK L:

This is the basic system clock. The falling edge of B CLK is used in specifying:

LD OSR L
USTK OUT ENABLE L
DISABLE HI NEXT H
ZERO HI NEXT L

2.2.6 MSEQ INIT L:

This is an initialization signal which is used throughout the machine.

The Internal MSEQ INIT signal is formed using a D type flip flop which is clocked with the rising edge of M CLK L. The input data to the flip flop is MSEQ INIT L. The flipflop is set if MSEQ INIT L is asserted. Internal MSEQ INIT is asserted whenever MSEQ INIT L is asserted or the flip flop is set. The internal signal is used to direct clear the USP and to force CS ADDR <Ø5:ØØ> = Ø

2.2.7 BUT CONTROL CODE H:

This signal is used in forming:

CS ADDR<Ø5:ØØ> L,
USTK ADDR<3:Ø>,
LD OSR L
USTK OUT ENABLE L
DISABLE HI NEXT H
ENABLE IRD ROM H
and in controlling USP<3:Ø>

It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.8 BUT <2:Ø> H:

These signals are used in forming:

CS ADDR<Ø5:ØØ> L,
USTK ADDR<3:Ø>,
LD OSR L
USTK OUT ENABLE L
DISABLE HI NEXT H
ENABLE IRD ROM H
and in controlling USP<3:Ø>
They are received through skew latches clocked with M CLK L and it is the latched versions that are actually used throughout this specification.

2.2.9 MICRO ADDR INH L:

When this signal is asserted CS ADDR <Ø5:ØØ> = Ø. It is also used to create:

ZERO HI NEXT L
USTK OUT ENABLE L
DISABLE HI NEXT H
ENABLE IRD ROM H

It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.1Ø ENABLE UVECT H:

This signal is used to form CS ADDR <Ø5:ØØ> L.

If ENABLE UVECT H or DO SRVC L is asserted:

USP <3:Ø> = USP <3:Ø> + 1
USTK ADDR <3:Ø> = USP <3:Ø> + 1
LD OSR L is not asserted
ZERO HI NEXT L is asserted
DISABLE HI NEXT H is asserted
ENABLE IRD ROM H is not asserted
USTK OUT ENABLE L is not asserted

It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.11 DO SRVC L:

See 2.2.1Ø.

2.2.12 IRD CTR <2:1> H:

This signal is used to form

CS ADDR <05:00> L,
USTK OUT ENABLE L
DISABLE MI NEXT H
ENABLE IRD ROM H
USTK ADDR <3:0> H
LD OSR L
and to control USP <3:0>

They are received through skew latches clocked with M CLK L and it is the latched versions that are actually used throughout this specification.

2.2.13 USTK ADDR <3:0> H:

These signals are used to addresr the micro-stack chips during stack operations. If JSR H, ENABLE UVECT H, or DO SRVC L is asserted then

USTK ADDR <3:0> = USP <3:0> + 1

IF the conditions described in the previous paragraph are not true, BUT CONTROL CODE H is asserted and (BUT<2:0> = 2,3 or (BUT<2:0> = 1 and IRD CTR<2:1> ≠ 0)) then

USTK ADDR <3:0> = USP <3:0> -1

Otherwise
USTK ADDR <3:0> = USP <3:0>.

2.2.14 USTK OUT ENABLE L:

This is the signal which controls the tri-state enable of the output of the micro-stack chips.

From the rising edge of M CLK L until the falling edge of the next B CLK L, the signal is inhibited. After the falling edge of B CLK L it is asserted if BUT CONTROL CODE H is asserted and (BUT<2:0> = 2,3 or (BUT<2:0> = 1 and IRD CTR<2:1> ≠0)). If

ENABLE UVECT H,
DO SRVC L
JSR H
Internal MSEQ INIT L
or MICRO ADDR INH L

are asserted then USTK OUT ENABLE L is inhibited.

2.2.15 JSR H:

This signal is used to form

USTK OUT ENABLE L
USTK ADDR<3:0> H
and to control USP<3:0>

It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.16 ZERO HI NEXT L:

This signal is used to clear the CS ADDR<13:06> lines external to MSQ. From the rising edge of M CLK L until the falling edge of the next B CLK L, the signal is inhibited. After the falling edge of B CLK L it is asserted when any of the following signals are asserted

Internal MSEQ INIT L
ENABLE UVECT H
DO SRVC L

If MICRO ADDR INH L is asserted, this signal is inhibited.

2.2.17 DISABLE HI NEXT H:

This signal is used to control the NEXT<13:06> Control Store latches. From the rising edge of M CLK L until the falling edge of the next B CLK L, the signal is asserted. After the falling edge of B CLK L it is asserted by any of the following conditions:

MICRO ADDR INH L is asserted
Internal MSEQ INIT L is asserted
ENABLE UVECT H is asserted
DO SRVC L is asserted
BUT CONTROL CODE H is asserted,
(BUT<2:0> = 2,3 or (BUT<2:0> = 1 and IRD CTR<2:1> ≠ 0)) and JSR H is not asserted.
BUT CONTROL CODE H is asserted and
(BUT<2:0> = 4 or (BUT<2:0> = 1 and IRD CTR<2:1> = 0))

Otherwise it is inhibited.

2.2.18 ENABLE IRD ROM H:

This signal is asserted when BUT CONTROL CODE H is asserted and (BUT<2:0> = 4 or (BUT<2:0> = 1 and IRD CTR<2:1> = 0)). If any of the following signals is asserted then ENABLE IRD ROM H is inhibited:

MICRO ADDR INH L
Internal MSEQ INIT L
ENABLE UVECT H
DO SRVC L

2.2.19 DISABLE OSR BUT H:

This signal comes from the IRD ROMs and when asserted inhibits LD OSR L.

2.2.20 LD OSR L:

The state of LD OSR L is initially determined during the interval between the rising edge of M CLK L and the next falling edge of B CLK L. Its value at that time is shown below.

BUT CONTROL CODE H	BUT <2:0> H	IRD CTR <2:1> H	LD OSR L
H	4,5,6,7	X	L
H	1	0	L
OTHERWISE			H

When the falling edge of B CLK L occurs until the rising edge of M CLK L, LD OSR L is generated as follows:

			DISABLE			
BUT CONTROL		IRD	OSR	DO SRVC	ENABLE	
CODE H	BUT<2:0> H	CTR<2:1> H	BUT L	L	UVECT H	LD OSR L
X	X	X	X	X	H	H
X	X	X	X	L	X	H
H	5,6,7	X	X	H	L	L
H	4	X	H	H	L	L
H	4	X	L	H	L	H
H	1	0	H	H	L	L
H	1	0	L	H	L	H
	Otherwise					H

2.2.21 LIT <1:0> H:

These signals are used to create FPA WAIT L.

2.2.22 FPA WAIT L:

This signal is asserted when LIT <1:0> = 2

MSQ TABLE 2

CS ADDR DATA (SEE PARAGRAPH 2.2.1)

	BUT	BUT	IRD	MICRO	ENABLE	DO	Internal	CS
JSR	CONTROL	<2:0>	CTR	ADDR	UVECT	SRVC	MSEQ	ADDR
H	CODE H	H	<2:1> H	INH L	H	L	INIT L	<05:00>
X	X	X	X	L	X	X	X	0
X	X	X	X	X	X	X	L	0
X	X	X	X	H	H	H	H	20 (HEX)
X	X	X	X	H	L	L	H	10 (HEX)
X	X	X	X	H	H	L	H	30 (HEX)
X	H	4	X	H	L	H	H	0
X	H	1	0	H	L	H	H	0
0	H	2or3	X	H	L	H	H	USTK D<05:00> + NEXT<05:00>
0	H	1	1,2,or3	H	L	H	H	USTK D<05:00> + NEXT<05:00>
Otherwise				H	L	H	H	NEXT<05:00>

```
                       -------------
         LIT 1 H --->!01         48!---> DISABLE HI NEXT H
        LD OSR L <---!02         47!<--- B CLK L
BUT CONTROL CODE H --->!03         46!<--- DO SRVC L
         BUT 2 H --->!04         45!<--- MSEQ INIT L
         BUT 0 H --->!05         44!<--- M CLK L
         BUT 1 H --->!06         43!<--- USTK D05 H
         LIT 0 H --->!07         42!<--- DISABLE OSR BUT L
      FPA WAIT L <---!08         41!---> ZERO HI NEXT L
  USTK ADDR 3 H <---!09         40!<--- MICRO ADDR INH L
  USTK ADDR 1 H <---!10 ....... 39!---> ENABLE IRD ROM H
  USTK ADDR 2 H <---!11 .     . 38!---- GROUND
             VGA ----!12 . LID . 37!---> USTK OUT ENABLE L
             VCC ----!13 . DOWN. 36!<--- JSR H
  USTK ADDR 0 H <---!14 .     . 35!---- GROUND
     IRD CTR 1 H --->!15 ....... 34!---> CS ADDR 05 L
     IRD CTR 2 H --->!16         33!<--- NEXT 05 H
      USTK D00 H --->!17         32!<--- NEXT 04 H
      USTK D01 H --->!18         31!---> CS ADDR 04 L
       NEXT 01 H --->!19         30!<--- USTK D03 H
    CS ADDR 00 L <---!20         29!<--- USTK D04 H
    CS ADDR 01 L <---!21         28!<--- NEXT 03 H
      USTK D02 H --->!22         27!---> CS ADDR 03 L
       NEXT 02 H --->!23         26!<--- ENABLE UVECT H
       NEXT 00 H --->!24         25!---> CS ADDR 02 L
                       -------------
```

MSQ FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.20 PROCESSOR STATUS BRANCHES (PHB) DC629

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for VAX-11/750 conditional micro-branches and processor registers.

PHB TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	C CLK ENABLE H		GA1TNF
2	CS ADDR 00 L	I_{OL}=12MA	GA1TCN
3	WBUS 00 H		GA1TZF
5	PSL FPD H		GA1TTN
6	WBUS 03 H		GA1TZF
7	INTERRUPT H		GA1TNF
8	CS ADDR 03 L	I_{OL}=12MA	GA1TCN
9	CS ADDR 02 L	I_{OL}=12MA	GA1TCN
10	DO SRVC L		GA1TNF
11	WBUS 01 H		GA1TZF
14	WBUS 02 H		GA1TZF
15	DISABLE CS ADD H		GA1TNF
16	BUT 0 H		GA1TNF
17	BUT 1 H		GA1TNF
18	INH BUT DECODE L		GA1TNF
19	BUT 4 H		GA1TNF
20	BUT 5 H		GA1TNF
21	BUT 2 H		GA1TNF
22	BUT 3 H		GA1TNF
23	WBUS 05 H		GA1TZF
24	CS ADDR 05 L	I_{OL}=12MA	GA1TCN
25	MISC CTL 2 H		GA1TNF
26	MISC CTL 4 H		GA1TNF
27	MISC CTL 1 H		GA1TNF
28	MISC CTL 3 H		GA1TNF
29	IRD ADD CTL 0 H		GA1TTN
30	LD OSR L		GA1TNF
31	IRD ADD CTL 1 H		GA1TTN
32	IRD LD RNUM H		GA1TTN
33	CS ADDR 04 L	I_{OL}=12MA	GA1TCN
34	PSL CM H		GA1TTN
36	WBUS 30 H		GA1TZF
37	PSL TP H		GA1TTN
39	WBUS 31 H		GA1TZF
40	WBUS 27 H		GA1TZF
41	PHB GD SAM 0 H		GA1TNF
42	PHB GD SAM 2 H		GA1TNF
43	WBUS 04 H		GA1TZF
44	PHB GD SAM 1 H		GA1TNF
45	M CLK L		GA1TNF
46	LD IR L		GA1TNF
47	MISC CTL 0 H		GA1TNF
48	CS ADDR 01 L	I_{OL}=12MA	GA1TCN

2. Performance:

2.1 State Elements:

2.1.1 Status Flag <5:0>:

These 6 bits are built from D Type flip flops. They are clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. The input data is as follows:

INH BUT		PHB	MISC						
DE-	BUT	GD	CTL						
CODE	<5:0>	SAM	<4:0>			STATUS FLAG DATA			
L	H	<2:0> H	H	5	4	3	2	1	0
H	4or5	X	X	0	0	0	0	0	0
(otherwise)		*5	6,7,0E,0F 10 - 1F	WBUS5	WBUS4	WBUS3	WBUS2	WBUS1	WBUS0
		0-4,6,7	0	-	-	-	-	-	0
			1	-	-	-	-	0	-
			2	-	-	-	0	-	-
			3	-	-	0	-	-	-
			4	-	0	-	-	-	-
			5	0	-	-	-	-	-
			8	-	-	-	-	-	1
			9	-	-	-	-	1	-
			0A	-	-	-	1	-	-
			0B	-	-	1	-	-	-
			0C	-	1	-	-	-	-
			0D	1	-	-	-	-	-
	Otherwise	Otherwise				No Change			

*See Note 1

Note 1: As far as micro-code use is concerned, when PHB GD SAM <2:0> = 5, MISC CTL codes 0-5 and 8-0D should not be specified. In reality however, if two different data sources are specified at once, the input data to the appropriate Status Flag is the OR of the 2 sources.

2.1.2 Step CTR <4:0>:

This is a 5 bit register built from D Type flip flops. It is clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. The input data is as follows:

INH BUT DECODE L	BUT <5:0> H	PHB GD SAM <2:0> H	MISC CTL <4:0> H	STEP CTR <4:0> DATA
H	4,5	X	X	0
H	*0C	0-5, 7	0-F, 10-12 18 - 1F	STEP CTR - 1
H	*0D	0-5, 7	|	STEP CTR - 4
(OTHERWISE)		*6	|	WBUS <4:0>
|		0-5, 7	13	STEP CTR - 1
|		|	14	2
|		|	15	6
|		|	16	14 (Decimal)
|		|	17	3C (Decimal)
		(OTHERWISE)		NO CHANGE

* As far as micro-code use is concerned, combinations of PHB GD SAM <2:0> = 6, MISC CTL codes 13-17, and BUT <5:0> = 0C or 0D should not be specified at the same time. In reality, however if two different data sources are specified at once, the input data to each Step Counter bit is the OR of the sources.

2.1.3 PSL CM:

This is a bit in the PSL which is a D Type flip flop. It is clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. If PHB GD SAM <2:0> = 7 then the input data is WBUS 31 H. Otherwise there is no change to the flip flop.

2.1.4 PSL FPD:

This is a bit in the PSL which is a D Type flip flop. It is clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. The input data is as follows:

PHB GD SAM <2:0>	MISC CTL <4:0>	INPUT DATA SOURCE
*7	0-1B, 1E, 1F	WBUS 27
*0 - 6	1C	0
*0 - 6	1D	1
OTHERWISE		NO CHANGE

* As far as micro-code use is concerned, when PHB GD SAM <2:0> = 7, MISC CTL codes 1C and 1D should not be specified. In reality however, if two different data sources are specified at once, the input data to the flip flop is the OR of the two sources.

2.1.5 PSL TBIT:

This is a bit in the PSL which is built from a D Type flip flop. It is clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. If PHB GD SAM <2:0> = 4 or 7 the input data is WBUS 04 H. Otherwise there is no change to the state of the flip flop.

2.1.6 BUT SERVICE Flip Flop:

This is a state element used to generate PSL TP H. It is built from a D Type flip flop and is clocked with the rising edge of M CLK L. The input data is LD IR L. When LD IR L is asserted the flipflop is set.

2.1.7 PSL TP:

This is a bit in the PSL which is a D Type flip flop. It is clocked with the rising edge of M CLK L. The input data is as follows:

BUT SERVICE FLIP FLOP	DO SRVC L	D CLK ENABLE H	PHB GD SAM <2:0>	MISC CTL <4:0>	INPUT DATA SOURCE
SET	H	H	* 0-6	X	PSL TBIT
SET	H	L	X	X	PSL TBIT
SET	L	H	* 0-6	0-F,10,11,13-1F	NO CHANGE
SET	L	L	X	X	NO CHANGE
CLEAR	X	L	X	X	NO CHANGE
CLEAR	X	H	* 7	0-F,10,11,13-1F	WBUS 30
CLEAR	X	H	* 0-6	12	0
OTHERWISE					NO CHANGE

*See Note 1.

Note 1: As far as micro-code use is concerned, when PHB GD SAM <2:0> = 7, MISC CTL code 12 should not be specified. In reality however, if the two micro-orders are specified at once, the input data to the flip flop is the OR of the two sources. If PHB GD SAM <2:0> = 7 while the BUT SERVICE flip flop is set, DO SRVC L is not asserted, and D CLK ENABLE H is asserted then the input data is the OR of PSL TBIT and WBUS 30. Also if MISC CTL <4:0> = 12 while the BUT SERVICE flip flop is set, DO SRVC L is not asserted and D CLK ENABLE H is asserted, the input data is PSL TBIT.

2.1.8 FIRST IRDX Flip Flop:

This is a state element used to generate IRD ADD CTL <1:0> H and IRD LD RNUM H. It is a D Type flip flop which is clocked with the rising edge of M CLK L whenever D CLK ENABLE H is asserted. The input data is as follows:

INH BUT DECODE L	BUT <5:0>	INPUT DATA SOURCE
H	4,5	0
H	1	1
OTHERWISE		NO CHANGE

2.2 Pin Descriptions:

Certain input pins are defined as having "Skew Latches". These are feed through latches that are open when the clock is not asserted and closed when the clock is asserted.

2.2.1 M CLK L:

This is the clock associated with the micro-sequencer. M CLK L is used to latch all state elements in PHB.

2.2.2 D CLK ENABLE H:

This signal is used to enable M CLK L when clocking:

STATUS FLAG <5:0>
STEP CTR <4:0>
PSL CM
PSL FPD
PSL TBIT
FIRST IRDX flip flop

2.2.3 LD IR L:

This input signal is used to load the BUT SERVICE flip flop. It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.4 DO SRVC L:

This signal is used to generate PSL TP H. It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.5 PHB GD SAM <2:0> H:

These signals specify the function to be performed by the WBUS lines used by PHB. They select the state elements:

STATUS FLAG <5:0>
STEP CTR <4:0>
PSL CM
PSL TP
PSL FPD
PSL TBIT

and control the reading and loading of the appropriate flip flops. See Paragraph 2.2.12. They are received through skew latches clocked with M CLK L and it is the latched versions that are actually used throughout this specification.

2.2.6 MISC CTL <4:0> H:

These signals are a field in the micro-word used to control various state elements. They are used here to generate:

STATUS FLAG <5:0>
STEP CTR <4:0>
PSL FPD
PSL TP
IRD LD RNUM H

They are received through skew latches clocked with M CLK L and it is the latched versions that are actually used throughout this specification.

2.2.7 BUT <5:0> H:

These signals make up the branch field in the micro-word. They are used to generate:

STATUS FLAG <5:0>
STEP CTR <4:0>
FIRST IRDX flip flop
CS ADDR <05:00> L
IRD ADD CTL <1:0> H
IRD LD RNUM H

They are received through skew latches clocked with M CLK L and it is the latched versions that are actually used throughout this specification.

2.2.8 INH BUT DECODE L:

This signal indicates that no decoding of the BUT field should be done. It is used wherever BUT <5:0> H are used.

It is received through a skew latch clocked with M CLK L and it is the latched version that is actually used throughout this specification.

2.2.9 DISABLE CS ADD H:

This signal is used to inhibit the effects of branch conditions in forming the CS Address. It is included in the generation of CS ADDR <05:00> L.

2.2.10 LD OSR L:

This signal specifies when the OSR is being loaded from the XB and is used to generate IRD ADD CTL <1:0> H and IRD LD RNUM H.

2.2.11 INTERRUPT H:

This signal is used for branching and is included in the generation of CS ADDR <05:00> L.

2.2.12 WBUS <31,30,27> H
WBUS <05:00> H

These tri-state lines are part of the WBUS which is used for reading and writing the state elements in the chip. They are controlled by PHB GD SAM <2:0> as follows:

PHB GD SAM <2:0>	WBUS USE	31	30	27	05	WBUS 04	03	02	01	00
0	NONE	-	-	-	-	Z (HIGH IMPEDANCE) -	-	-	-	-
1	OUTPUT	PSL CM	PSL TP	PSL FPD	-	STATUS FLAG <5:0>	-	-	-	
2	OUTPUT	PSL CM	PSL TP	PSL FPD	Z	PSL TBIT	Z	Z	Z	Z
3	OUTPUT	PSL CM	PSL TP	PSL FPD	STATUS FLAG 5	-	-	STEP CTR <4:0>		-
4	INPUT	Data to be loaded into PSL TBIT								
5	INPUT	Data to be loaded into STATUS FLAG <5:0>								
6	INPUT	Data to be loaded into STEP CTR <4:0>								
7	INPUT	Data to be loaded into PSL CM, PSL TP, PSL FPD and PSL TBIT								

The exact WBUS bits used as input for the various registers is described in Paragraph 2.1.

2.2.13 CS ADDR <05:00> L:

These signals are used to generate branch condition offsets. Outside the chip they are ORed with other sources of CS ADDR <13:00> L to form a destination address. They are sourced by PHB as shown in PHB Chart 1.

2.2.14 PSL CM H:

This signal is the output of the PSL CM flip flop.

2.2.15 PSL TP H:

This signal is the output of the PSL TP flip flop.

2.2.16 PSL FPD H:

This signal is the output of the PSL FPD flip flop.

DIS-	INH							
ABLE	BUT							
CS	DE-	BUT						
ADD	CODE	<5:0>			CS ADDR <05:00> H			
H	L	H	05	04	03	02	01	00
H	X	X	0	0	0	0	0	0
X	L	X	0	0	0	0	0	0
L	H	8	WBUS05	WBUS04	WBUS03	WBUS02	WBUS01	WBUS00
L	H	9	0	0	0	0	WBUS01	WBUS00
L	H	A	0	0	0	0	0	WBUS00
L	H	B	0	0	0	0	.NOT.STATUS FLAG 4 AND INTERRUPT	0
L	H	C	0	0	0	0	0	STEP CTR - 1 = 0
L	H	D	0	0	0	STEP CTR >=4	STEP CTR = 2 OR 3	STEP CTR = 1 OR 3 OR STEP CTR >= 4 AND INTERRUPT
L	H	E	0	0	0	0	0	WBUS <1:0> ≠ 0
L	H	F	0	0	0	0	0	PSL FPD
L	H	10	0	0	0	0	0	STATUS FLAG 0
L	H	11	0	0	0	0	0	STATUS FLAG 4
L	H	12	0	0	0	STATUS FLAG 2	0	0
L	H	13	0	0	0	0	0	STATUS FLAG 3
L	H	14	0	0	0	0	STATUS FLAG 1	0
L	H	15	0	0	0	0	STATUS FLAG 1	STATUS FLAG 2 XOR STATUS FLAG 3
L	H	16	0	0	0	STATUS FLAG 2	STATUS FLAG 1	STATUS FLAG 0
L	H	17	0	0	0	0	STATUS FLAG 1	0
L	H	18	0	0	0	0	0	0
L	H	19	0	0	0	0	0	0
L	H	1A	0	0	0	0	0	STATUS FLAG 5
L	H	1B	0	0	0	0	WBUS 31	WBUS 30
OTHERWISE			0	0	0	0	0	0

PHB CHART 1

2.2.17 IRD ADD CTL <1:0> H:

These signals indicate whether branching is to be done on operand specific information. They are generated as follows:

DIS- ABLE CS ADD H	INH BUT DE- CODE L	BUT <5:0>	PSL CM FLIP FLOP	FIRST IRDX FLIP FLOP	LD OSR L	IRD ADD CTL <1:0> H
H	X	X	X	X	X	0
L	H	NOT 5,18,19	CLEAR	X	L	1
L	H	4 OR 5	SET	X	X	1
L	H	18	SET	X	X	2
L	H	18	CLEAR	X	H	2
L	H	1	SET	CLEAR	X	2
L	H	19	X	X	X	3
L	L	X	CLEAR	X	L	1
L	H	18	CLEAR	X	L	3
OTHERWISE						0

2.2.18 IRD LD RNUM H:

This signal specifies when the RNUM register should be loaded. It is generated as follows:

D CLK ENABLE H	MISC CTL <4:0> H	PSL CM FLIP FLOP	FIRST IRDX FLIP FLOP	INH BUT DECODE L	BUT <5:0>	LD OSR L	IRD LD RNUM H
X	X	X	X	X	X	L	H
L	X	X	X	X	X	H	L
H	11	X	X	X	X	H	H
H	NOT 11	X	X	L	X	H	L
H	NOT 11	X	X	H	NOT 1	H	L
H	|	CLEAR	X	H	1	H	L
H	|	SET	CLEAR	H	1	H	H
H	|	SET	SET	H	1	H	L
	OTHERWISE						L

```
                                   --------------
      D CLK ENABLE H --->!01          48!---> CS ADDR 01 L
        CS ADDR 00 L <---!02          47!<--- MISC CTL 0 H
           WBUS 00 H <-->!03          46!<--- LD IR L
                      NC !04          45!<--- M CLK L
           PSL FPD H <---!05          44!<--- PHB GD SAM 1 H
           WBUS 03 H <-->!06          43!<--> WBUS 04 H
         INTERRUPT H --->!07          42!<--- PHB GD SAM 2 H
        CS ADDR 03 L <---!08          41!<--- PHB GD SAM 0 H
        CS ADDR 02 L <---!09          40!<--> WBUS 27 H
           DO SRVC L --->!10 ....... 39!<--> WBUS 31 H
           WBUS 01 H <-->!11 .     . 38!---- GROUND
                 VGA ----!12 . LID . 37!---> PSL TP H
                 VCC ----!13 . DOWN. 36!<--> WBUS 30 H
           WBUS 02 H <-->!14 .     . 35!---- GROUND
 DISABLE CS ADD H --->!15 ....... 34!---> PSL CM H
             BUT 0 H --->!16          33!---> CS ADDR 04 L
             BUT 1 H --->!17          32!---> IRD LD RNUM H
  INH BUT DECODE L --->!18          31!---> IRD ADD CTL 1 H
             BUT 4 H --->!19          30!<--- LD OSR L
             BUT 5 H --->!20          29!---> IRD ADD CTL 0 H
             BUT 2 H --->!21          28!<--- MISC CTL 3 H
             BUT 3 H --->!22          27!<--- MISC CTL 1 H
           WBUS 05 H <-->!23          26!<--- MISC CTL 4 H
        CS ADDR 05 L <---!24          25!<--- MISC CTL 2 H
                                   --------------
```

PHB FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.21 PREFETCH CONTROLLER CHIP (PRK-DC624)

1. GENERAL DESCRIPTION:

PRK is a control chip which monitors usage of memory interface resources and stalls the CPU when it detects an attempt to use a resource which is already busy. In addition, PRK controls I-stream prefetching and generates several control signals for the ADDRESS chip bit slices.

PRK TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	LATCHED MSRC 1 H		GA1TNF
2	SNAPSHOT CMI L		GA1TNF
3	B CLK L		GA1TNF
4	PHASE 1 H		GA1TNF
5	MMUX SEL S1 H		GA1TTN
6	ENABLE ACV STALL H		GA1TTN
7	MSEQ INIT L		GA1TNF
8	STALL L		GA1TTN
9	LATCH MA L		GA1TTN
10	PREFETCH L		GA1TTN
11	ENA VA SAVE L		GA1TTN
14	UTRAP L		GA1TNF
15	M CLK ENABLE H		GA1TNF
16	D CLK ENABLE H		GA1TNF
17	LATCHED BUS 3 H		GA1TNF
18	STATUS VALID L		GA1TNF
19	LATCHED WCTRL 1 H		GA1TNF
20	XB1 IN USE L		GA1TTN
21	XB0 IN USE L		GA1TTN
22	LATCHED WCTRL 3 H		GA1TNF
23	XB SELECT H		GA1TTN
24	ENA PC L		GA1TTN
25	LATCHED WCTRL 4 H		GA1TNF
26	LATCHED WCTRL 5 H		GA1TNF
27	LATCHED WCTRL 2 H		GA1TNF
28	LATCHED WCTRL 0 H		GA1TNF
29	BUS 4 H		GA1TNF
30	XB PC 01 H		GA1TNF
31	LATCHED BUS 2 H		GA1TNF
32	PSL CM H		GA1TNF
33	LATCHED BUS 1 H		GA1TNF
34	LATCHED BUS 0 H		GA1TNF
36	DST RMODE H		GA1TNF
37	XB PC 00 H		GA1TNF
39	I SIZE 1 L		GA1TNF
40	LD OSR L		GA1TNF
41	I SIZE 0 L		GA1TNF
42	IRD1 H		GA1TNF
43	MA SELECT S0 H		GA1TTF
44	LATCHED MSRC 4 H		GA1TNF
45	LATCHED MSRC 0 H		GA1TNF
46	MA SELECT S1 H		GA1TTF
47	LATCHED MSRC 3 H		GA1TNF
48	LATCHED MSRC 2 H		GA1TNF

2. **Performance:**

PRK I/O PINS

INPUTS	#PINS
B CLK L	1
BUS 4 H	1
D CLK ENABLE H	1
DST RMODE H	1
I SIZE <1:Ø> L	2
IRD1 H	1
LATCHED BUS <3:Ø> H	4
LATCHED MSRC <4:Ø> H	5
LATCHED WCTRL <5:Ø> H	6
LD OSR L	1
M CLK ENABLE H	1
MSEQ INIT L	1
PHASE 1 H	1
PSL CM H	1
SNAPSHOT CMI L	1
STATUS VALID L	1
UTRAP L	1
XB PC <Ø1:ØØ> H	2

OUTPUTS (TOTEM POLE)	
ENA PC L	1
ENA VA SAVE L	1
ENABLE ACV STALL H	1
LATCH MA L	1
MA SELECT S<1:Ø> H	2
MMUX SEL S1 H	1
PREFETCH L	1
STALL L	1
XB SELECT H	1
XB<1:Ø> IN USE L	2

2.1 The following MSRC codes are of interest in the MEMORY INTERFACE:

(HEX)		
	12.	SOURCE MDR
	13.	SOURCE WDR
	17.	SOURCE XB
	18.	SOURCE MA
	19.	SOURCE PC SAVE
	1A.	SOURCE PC
	1B.	SOURCE VA
	1F.	SOURCE TB DATA

MSRC Functions are decoded from "LATCHED MSRC <4:0> H"

2.2 The following is a list of the code assignments for the BUS FUNCTION Micro field:

0. READ PHYSICAL ADDRESS
1. PROCESSOR INITIALIZE
2. READ, NO MICRO-TRAP
3. I/O INITIALIZE
4. READ LOCK TIMEOUT TEST
5. NOP
6. READ, SECOND REFERENCE
7. NOP
8. WRITE PHYSICAL ADDRESS
9. REI CHECK

A. WRITE, SECOND REFERENCE
B. WRITE UNLOCK, SECOND REFERENCE
C. WRITE, NO MICRO-TRAP
D. NOP
E. WRITE LONGWORD, NO MICRO-TRAP
F. BUS GRANT
10. READ
11. READ LONGWORD
12. PTE ACCESS CHECK, WRITE
13. READ LOCK
14. READ WITH MODIFY INTENT
15. READ LONGWORD WITH MODIFY INTENT
16. PTE ACCESS CHECK, READ
17. PTE ACCESS CHECK, READ, KERNEL MODE
18. WRITE
19. WRITE LONGWORD
1A. WRITE IF NOT RMODE
1B. WRITE UNLOCK
1C. PROBE ACCESS, WRITE, MODE SPECIFIED
1D. PROBE ACCESS, WRITE
1E. PROBE ACCESS, READ, MODE SPECIFIED
1F. PROBE ACCESS, READ

Bus Functions are decoded from "LATCHED BUS 4 H"(FLIP-FLOP) and "LATCHED BUS <3:0> H"

2.3 The following is a list of the code assignments for the upper half of the WCTRL Micro field:

20. VA <- PC + ISIZE + (WBUS)
    PC <- PC + ISIZE
21. RESERVED
22. VA <- VA + 4
23. MDR <- (WBUS)
24. PC <- (WBUS)
25. VA <- (WBUS)
26. MBUS <- WDR
27. MDR <- 0
28. TB DATA <- (WBUS)

29. TB VALID BIT <- 0
    VA <- (WBUS)
    (Invalidate both groups at the index position addressed by VA).
2A. WDR <- (WBUS) UNROTATED
2B. MDR <- IR, ZERO EXTENDED
2C. PC <- PC + (WBUS)
2D. CACHE VALID BIT <- 0
    VA <- (WBUS)
    (Invalidate both groups at the index position addressed by VA. The address in the VA register will be interpreted as a physical address).
2E. WDR <- (WBUS)
2F. MDR <- OSR, ZERO EXTENDED
30. STATUS/CONTROL REGISTER <- WBUS<27:24>
31. PREVIOUS MODE REGISTER <- WBUS<23:22>
32. WBUS<27:24> <- STATUS/CONTROL REGISTER
33. BUS GRANT
    WBUS<20:16> <- IPL OF CURRENT UNIBUS GRANTEE
34. STATUS/CONTROL ADDRESS REGISTER <- WBUS<27:24>
35. PREVIOUS MODE REGISTER <- CURRENT MODE REGISTER, THEN IS/CURRENT MODE REGISTER <- WBUS<26:24>
37. REI CHECK
38. ASTLVL kEGISTER <- WBUS<26:24>
39. (RESERVED)
3A. WBUS<26:24> <- ASTLVL REGISTER
3B. (RESERVED)
3C. HIGHEST SOFTWARE IPR REGISTER <- WBUS<20:16>
3D. IPL REGISTER <- WBUS<20:16>
3E. RESERVED
3F. WBUS<20:16> <- IPL OF LAST UNIBUS GRANTEE

WCTRL Functions are decoded from "LATCHED WCTRL <5:0> H"

2.4 Internal Signal Definitions:

#"LATCHED BUS 4 H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"M CLK ENABLE H" & "BUS 4 H"

K INPUT:

"M CLK ENABLE H" & "BUS 4 L"

*"BUS CYC DEC H" is TRUE for any of the following Bus Function Decodes:

READ PHYSICAL ADDRESS
READ, NO MICRO-TRAP
READ LOCK TIMEOUT TEST
READ, SECOND REFERENCE
WRITE PHYSICAL ADDRESS
WRITE, SECOND REFERENCE

WRITE UNLOCK, SECOND REFERENCE
WRITE, NO MICRO-TRAP
WRITE LONGWORD, NO MICRO-TRAP
BUS GRANT
READ
READ LONGWORD
READ LOCK
READ WITH MODIFY INTENT
READ LONGWORD WITH MODIFY INTENT
WRITE
WRITE LONGWORD
WRITE UNLOCK

OR

WRITE IF NOT RMODE & "DST RMODE L"

*"BUS GRANT DEC H" is TRUE for: (Bus Function Decode)

BUS GRANT

*"BUS REQ H" is TRUE for: (WCTRL Decodes)

[23 + 26 + 27 + 28 + 29 + 2A + 2B + 2D + 2E + 2F]

OR

Any Bus Function Decode EXCEPT:

PROCESSOR INITIALIZE
I/O INITIALIZE
NOP

*"DEST MDR H" is TRUE for any of the following WCTRL Decodes:

MDR <- (WBUS)
MDR <- 0
MDR <- IR, ZERO EXTENDED
MDR <- OSR, ZERO EXTENDED

*"DEST WDR H" is TRUE for any of the following WCTRL Decodes:

WDR <- (WBUS)
WDR <- (WBUS) UNROTATED

*"MSRC XB H" is TRUE for: (MSRC Decode)

SOURCE XB

*"PA BUS REQ H" is TRUE for any of the following Bus Function Decodes:

```
   PTE ACCESS CHECK, WRITE
   PTE ACCESS CHECK, READ
   PTE ACCESS CHECK, READ, KERNEL MODE
   PROBE ACCESS, WRITE, MODE SPECIFIED
   PROBE ACCESS, WRITE
   PROBE ACCESS, READ, MODE SPECIFIED
   PROBE ACCESS, READ
```

OR any of the followinj WCTRL Decodes:

```
   TB DATA <- (WBUS)
   TB VALID BIT <- 0
   VA <- (WBUS)

   CACHE VALID BIT <- C
   VA <- (WBUS)
```

*"ENA LD OSR H" is the HIGH TRUE output of an RS FLIP-FLOP whicn is set by:

```
   ("B CLK H" & "PHASE 1 L")
```

and reset by:

```
   ("B CLK H" & "PHASE 1 H")
```

*"CLK OSR H" is the HIGH TURE output of a latch which is enabled during "ENA LD OSR H".

```
   "LD OSR H"
```

*"BYTES REQ 0 H" is TRUE if:

```
   ("I SIZE 0 H" & "MSRC XB H") +
   ["MSRC XB L" & ("LD OSR H" .xor. "IRD1 H")]
```

*"BYTES REQ 1 H" is TRUE if:

```
   ("I SIZE 1 H" & "MSRC XB H") +
   ("IRD1 H" & "LD OSR H" & "MSRC XB L")
```

*"BYTES REQ H" is TRUE if:

```
   "PHASE 1 H" & ("BYTES REQ 1 H" + "BYTES REQ 0 H)
```

*"BOTH XBS REQ H" is TRUE if:

```
   ("PC 01 H" & "BYTES REQ 1 H" & "BYTES REQ 0 H") +
   ("PC 00 H" & "BYTES REQ 1 H" & "BYTES REQ 0 H") +
   ("PC 01 H" & "PC 00 H" & "BYTES REQ 1 H")
```

*"DEST PC H" is TRUE if:

[24 + 2C](WCTRL Decodes)
*"PC ENABLE H" is TRUE if:

"UTRAP L" &
("BYTES REQ 1 H" + "bYTES REC 0 H" + "DEST PC H")

*"LOAD PC H" is TRUE if:

"DEST PC H" & "PC ENABLE H" & "M CLK ENABLE H"

#"PC CLK REQ H" - JK FLIP-FLOP (FALLING edge of "B CLK L") J INPUT:

"PHASE 1 H" & "PC ENABLE H"

K INPUT:

("PHASE 1 H" & "PC ENABLE L") +
("M CLK ENABLE H" & "CYC IN PROG H")

*"TOGGLE 2 H" is TRUE if:

("PC ENABLE H" & "M CLK ENABLE H") &
("PC 01 H" + "BYTES REQ 1 H") &
("PC 00 H" + "BYTES REQ 1 H") &
("PC 01 H" + "BYTES REQ 0 H") &
("BYTES REQ 1 H" + "BYTES REQ 0 H")

*"STEER COMP DUMP H" is TRUE if:

"BUS CYC DEC H" & "BUS GRANT DEC L" &
"LATCHED BUS 3 H" & "PSL CM H" & "D CLK ENABLE H"

*"STEER DUMP H" is TRUE if:

"LOAD PC H" + "STEER COMP DUMP H"

#"PHASE 2 DEL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"PHASE 1 L"

The flop is DC cleared when "PHASE 1 H" is TRUE.

#"LATCHED BUS 4 H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"M CLK ENABLE H" & "BUS 4 H"
K INPUT:

"M CLK ENABLE H" & "BUS 4 L"

#"LATCHED UTRAP H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"UTRAP H"

K INPUT:

"BYTES REQ H" + "LOAD PC H"

The flop is DC PRESET when "MSEQ INIT H" is TRUE.

#"PREFETCH DEL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"PREFETCH H"

The flop is DC cleared when "PREFETCH H" is FALSE.

#"PREFETCH CYC H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"PREFETCH DEL H"

The flop is DC PRESET when "PREFETCH DEL H" is TRUE.

#"STATUS VAL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"MSEQ INIT H"

The flop is DC PRESET if:

"B CLK L" & "STATUS VALID H" &
("MSEQ INIT H" + "STATUS VAL DEL L")

#"STATUS VAL DEL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"STATUS VAL H"

#"ABORTED CYC H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"PREFETCH DEL H" & "STEER DUMP H" & "STATUS VAL L"
K INPUT:

"STATUS VAL H" + "MSEQ INIT H"

#"FORCE BUS ADD H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"PREFETCH L" & "M CLK ENABLE H" & "BUS 4 H"

K INPUT:

"PREFETCH H" + ("M CLK ENABLE H" & "BUS 4 L")
*"STEER VA H" is TRUE if:

"BUS 4 H" & "M CLK ENABLE H" & "PREFETCH L"

*"MEM REQ H" is TRUE if:

"PREFETCH DEL H" +
("BUS CYC DEC H" & "PREFETCH L" & "REPLACEMENT L")

*"REPLACEMENT H" is TRUE if:

"STATUS VAL H" & "ADD ENA DEL L" & "READ H"

*"ENA MSRC ADD H" is TRUE if:

("PREFETCH L" & "PHASE 1 H") &
[SOURCE: MA + PC + PC SAVE + VA(MSRC Decodes)]

*"PREFETCH INH H" is TRUE if:

"LATCHED UTRAP H" & "PREFETCH DEL L" & "BYTES REQ L"

#"ADD REG ENA H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"MEM REQ H" & "INVAL CHECK L" & "CYC IN PROG L"

K INPUT:

"PREFETCH CYC H" + "M CLK ENABLE H"

#"ADD ENA DEL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"ADD REG ENA H"

The flop is DC set when "ADD REG ENA H" is TRUE.

*"CYC IN PROG H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

"ADD REG ENA H"

and reset by:

"STATUS VAL H"

*"READ H" is the HIGH TRUE output of a latch which is enabled during "ADD REG ENA H". LATCH Input is:

"PREFETCH H" + "LATCHED BUS 3 L"

*"READ CYC H" is TRUE if:
"PREFETCH H" + "REPLACEMENT H" +
("CYC IN PROG H" & "ADD REG ENA L" & READ H")

#"MMUX S1 H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

("SNAPSHOT CMI L" & "XB STALL L" & "LATCHED XB STALL L") & ["ENA MSRC ADD H" + ("CYC IN PROG L" & "PREFETCH L" & SOURCE TB DATA(MSRC Decode))]

The flop is DC CLEARED when "PHASE 1 H" is FALSE.

*"MMUX SEL S1 H" is TRUE if:

"MMUX S1 H" + ("PHASE 1 L" & "PA BUS REQ H")

*"RESET ADD ENA H" is TRUE if:

"PREFETCH CYC H" + "M CLK ENABLE H"

#"INVAL CHECK H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

("MMUX SEL S1 L" & "SNAPSHOT CMI H") & ["MEM REQ L" + ("ADD REG ENA L" & "CYC IN PROG H") + ("ADD REG ENA H" & "RESET ADD ENA H")]

K INPUT:

"SNAPSHOT CMI L"

#"INVAL WRITE H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

"INVAL CHECK H"

The flop is DC cleared when "INVAL CHECK H" is FALSE.

#"DELAY H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

("INVAL CHECK H" & "PA BUS REQ H")

*"LOAD MDR H" is TRUE if:

"STATUS VAL H" & "READ H" & "PREFETCH CYC L"

#"MDR LDD H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

"LOAD MDR H"

K INPUT:

"BUS CYC DEC H" & "LATCHED BUS 3 L" & "D CLK ENABLE H"

#"XB SEL DEL H" - D FLIP-FLOP (RISING edge of "B CLK L")

D INPUT:

"XB SEL H"

#"XB1 LDD H" - JK FLIP-FLOP (RISING edge of "B CLK L")
J INPUT:

"STATUS VAL H" & "PREFETCH CYC H" & "XB SEL DEL H" &
"ABORTED CYC L" & "LOAD PC L" & "STEER COMP DUMP L"

K INPUT:

"STEER COMP DUMP H" + "LOAD PC H" + "MSEQ INIT H" +
("TOGGLE 2 H" & "XB SEL L")

#"XB0 LDD H" - JK FLIP-FLOP (RISING edge of "B CLK L")
J INPUT:

"STATUS VAL H" & "PREFETCH CYC H" & "XB SEL DEL L" &
"ABORTED CYC L" & "LOAD PC L" & "STEER COMP DUMP L"

K INPUT:

"STEER COMP DUMP H" + "LOAD PC H" + "MSEQ INIT H" +
("TOGGLE 2 H" & "XB SEL H")

*"RESET XB1 LDD H" is the HIGH TRUE output of a latch which is enabled during "B CLK H". LATCH Input is:

("STEER COMP DUMP H" + "LOAD PC H") +

[("XB1 LDD H" + "ABORTED CYC H" + "XB SEL L" +
"PREFETCH CYC L" + "STATUS VAL L") &
("XB1 LDD L" + "MSEQ INIT H" +
"TOGGLE 2 H" & "XB SEL L")]

*"RESET XB0 LDD H" is the HIGH TRUE output of a latch which is enabled during "B CLK H". LATCH Input is:

("STEER COMP DUMP H" + "LOAD PC H") +

[("XB0 LDD H" + "ABORTED CYC H" + "XB SEL H" +
"PREFETCH CYC L" + "STATUS VAL L") &
("XB0 LDD L" + "MSEQ INIT H" +
"TOGGLE 2 H" & "XB SEL H")]

#"XB SEL H" - JK FLIP-FLOP (RISING edge of "B CLK L")
J INPUT:

"RESET XB1 LDD H" & "RESET XB0 LDD L"
K INPUT:

"RESET XB0 LDD H"
*"XB REQ H" is TRUE if:

```
   ("XB1 LDD L" + "XB SEL H" + "BOTH XBS REQ H") &
   ("XB0 LDD L" + "XB SEL L" + "BOTH XBS REQ H") &
   ("BYTES REQ 1 H" + "BYTES REQ 0 H") &
   ("XB1 LDD L" + "XB0 LDD L")

*"FILL XB REQ H" is TRUE if:

   ("STATUS VAL H" & "PREFETCH CYC H" & "ABORTED CYC L")
   & ("XB1 LDD H" + "XB0 LDD H")

*"PREFETCH REQ H" is TRUE if:

   ("XB1 LDD L" & "XB0 LDD L") +
   [("XB1 LDD L" + "XB0 LDD L") &
   ("PREFETCH CYC L" + "ABORTED CYC H")]

*"ENA PRE ADD H" is TRUE if:

   "TOGGLE 2 H" + "STEER COMP DUMP H" + "LOAD PC H" +
   ["PREFETCH REQ H" & ("M CLK ENABLE H" + "PREFETCH
   H")]

*"FORCE VA H" is TRUE if:

   ("ENA MSRC ADD L" & "ENA PRE ADD L") +
   ("ENA MSRC ADD H" & "LATCHED MSRC 1 L" & "LATCHED
   MSRC 0 L")

#"MA SEL S1 H" - D FLIP-FLOP (RISING edge of "B CLK L")
D INPUT:

   "STEER VA H" +

   "FORCE VA H" +

   ("ENA MSRC ADD H" & "LATCHED MSRC 1 H") +

   ["ENA PRE ADD H" &
   ("CYC IN PROG L" + "PREFETCH CYC L" + "ABORTED CYC
   H")] & ["TOGGLE 2 H" & ("XB1 LDD L" + "XB0 LDD L") +
   "STEER DUMP H" + ("XB0 LDD L" & "XB1 LDD L")]

#"MA SEL S0 H" - D FLIP-FLOP (RISING edge of "B CLK L")
D INPUT:

   "STEER VA H" + "FORCE VA H" +
   ("ENA MSRC ADD H" & "LATCHED MSRC 0 H")
#"LATCH MA H" - JK FLIP-FLOP (FALLING edge of "B CLK L")
J Input:

   "UTRAP H" +
   ["ENA MSRC ADD L" &
   ("MA SEL S1 H" + "PREFETCH H") &
   ("MA SEL S0 H" + "PREFETCH H") &
("INVAL CHECK L" + "INVAL WRITE H") &
("ADD REG ENA H" + "CYC IN PROG L")] &
[("PA BUS REQ H" & "PREFETCH L") + "MEM REQ H"]
```

K Input:

```
("UTRAP L" & "DEST PC H" &
"M CLK ENABLE H" & "LATCHED UTRAP H") +
("UTRAP L" & "LATCHED UTRAP L") &
[("M CLK ENABLE H" & "PREFETCH L") +
("PREFETCH CYC H" & "ADD REG ENA H")] +
("LATCHED UTRAP H" & "PHASE 1 H") &
("BYTES REQ H" + "PA BUS REQ H"
"BUS CYC DEC H" + "ENA MSRC ADD H")
```

The flop is CLEARED when "INIT H" is TRUE.

#"LATCH MA DEL H" - D FLIP-FLOP (RISING edge of "B CLK L") D INPUT:

```
"LATCH MA H"
```

#"TIM ACV STALL H" - JK FLIP-FLOP (RISING edge of "B CLK L") J INPUT:

```
"PHASE 1 L" & "ENABLE ACV STALL H"
```

K INPUT:

```
"PHASE 1 H"
```

*"ENABLE ACV STALL H" is TRUE if:

```
"PHASE 1 L" & "PREFETCH L" & "ADD REG ENA H" &
"LATCH MA DEL H" & "TIM ACV STALL L" & "BUS CYC DEC
H"
```

*"LATCHED XB STALL H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

```
"PREFETCH H" & "PREFETCH INH L" &
"FILL XB REQ L" & "XB REQ H" & "B CLK H"
```

and reset by:

```
"PREFETCH INH H" + "PREFETCH L" + "FILL XB REQ H"
```

*"XB STALL H" is TRUE if:

```
"PREFETCH INH L" & "FILL XB REQ L" & "XB REQ H"
```

*"DELAY STALL H" is TRUE if:

```
"DELAY H" & "PHASE 1 L"
```

*"BUS CYC STALL H" is TRUE if:

[("EUS CYC DEC H" & "ENA MSRC ADD L" & "PHASE 1 H") &
("MA SEL S1 L" + "MA SEL SØ L" + "PREFETCH H")] +
[("BUS CYC DEC H" & "PHASE 1 L") &
("ADD REG ENA L" + "PREFETCH H" + "LATCH MA DEL L")]

*"PA BUS PH1 STALL H" is TRUE if:

("PA BUS REQ H" & "ENA MSRC ADD L" & "PHASE 1 H") &

("MA SEL S1 L" + "MA SEL SØ L" + "PREFETCH H" +
"INVAL CHECK H" + "STATUS VAL H" + "CYC IN PROG H")

*"PA BUS PH2 STALL H" is TRUE if:

("PA BUS REQ H" & "PHASE 1 L") &
("INVAL CHECK H" + "LATCH MA DEL L")

*"MSRC STALL H" is TRUE if:

[("MMUX SEL S1 L" & "PHASE 1 H") &
(SOURCE: TB DATA + MA + PC + PC SAVE + VA(MSRC Decode))] +

["PHASE 1 H" & "READ CYC H" & SOURCE WDR(MSRC Decode)]

*"MDR STALL H" is TRUE if:

"LOAD MDR L" & "MDR LDD L" & SOURCE MDR(MSRC Decode)

*"MEM CYC STALL H" is TRUE if:

["READ CYC H" & "PHASE 1 L" &
("DEST MDR H" + "DEST WDR H")] +

[("DEST WDR H" & "PHASE 1 L") &
("MA SEL S1 L" + "MA SEL SØ L" +
"CYC IN PROG H" & "READ L" & "ADD REG ENA L")]

*"INIT A H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

"INIT H"

and reset by:

"STATUS VALID L"

*"NO PREFETCH H" is TRUE if:

("XB1 LDD H" & "XBØ LDD H") +

```
        [(SOURCE: MA + PC + PC SAVE + VA + TB DATA +
        XB + WDR(MSRC Decodes)) & "PHASE 1 H" &
        "XB STALL L" & "LATCHED XB STALL L" & "MSRC XB L" &
        ("STATUS VAL H" + "PREFETCH DEL L")] +

        ("CYC IN PROG H" & "PREFETCH CYC L") +

        [("BUS REQ H" & "XB STALL L" & "LATCHED XB STALL L")
        & ("STATUS VAL H" + "PREFETCH DEL L")] +

        ("UTRAP H" & "PREFETCH L")

     *"NO PREFETCH A H" is TRUE if:

        ("CYC IN PROG L" & "INVAL CHECK H" & "INVAL WRITE L")


        + ("CYC IN PROG L" & "PHASE 1 L" &
        "PC CLK REQ H" & "LATCH MA L") +

        ("BYTES REQ L" & "LATCHED UTRAP H" & "PREFETCH DEL
        L") +

        ["STATUS VAL H" & ("UTRAP H" + "PREFETCH REQ L")] +

        ("PREFETCH CYC H" & "ADD ENA DEL L" &
        "STATUS VAL H" & "READ H") +

        "INIT A H"

     *"PREFETCH H" is TRUE if:

        "NO PREFETCH L" & "NO PREFETCH A L"
```

CHIP OUTPUTS

```
    >>>  "ENA PC L" is TRUE (LOW) if:
         "M CLK ENABLE H" & "PC ENABLE H"

    >>>  "ENA VA SAVE L" is TRUE (LOW) if:

         "PHASE 1 H" + "DEST PC H"

    >>>  "ENABLE ACV STALL H" is TRUE (HIGH) if:

         "ENABLE ACV STALL H"

    >>>  "LATCH MA L" is TRUE (LOW) if:

         "LATCH MA H"

    >>>  "MA SELECT S1 H" is TRUE (HIGH) if:

         "MA SEL S1 H"
```

```
>>>    "MA SELECT S0 H" is TRUE (HIGH) if:

       "MA SEL S0 H"

>>>    "MMUX SEL S1 H" is TRUE (HIGH) if:

       "MMUX S1 H" + ("PHASE 1 L" & "PA BUS REQ H")

>>>    "PREFETCH L" is TRUE (LOW) if:

       "PREFETCH H"

>>>    "STALL L" is TRUE (LOW) if:

       "MSEQ INIT L" &
       ["DELAY STALL H" + "BUS CYC STALL H" + "XB STALL H" +
       "PA BUS PH1 STALL H" + "PA BUS PH2 STALL H" +
       "MSRC STALL H" + "MDR STALL H" + "MEM CYC STALL H"]

>>>    "XB SELECT H" is TRUE (HIGH) if:

       "XB SEL H"

>>>    "XB1 IN USE L" is TRUE (LOW) if:

       "RESET XB1 LDD L" & ("BYTES REQ 1 H" + "BYTES REQ 0
       H") & ("BOTH XBS REQ H" + "XB SEL L")

>>>    "XB0 IN USE L" is TRUE (LOW) if:

       "RESET XB0 LDD L" & ("BYTES REQ 1 H" + "BYTES REQ 0
       H") & ("BOTH XBS REQ H" + "XB SEL H")
```

```
                                         ---------------
    LATCHED MSRC 1 H --->!01                   48!<--- LATCHED MSRC 2 H
      SNAPSHOT CMI L --->!02                   47!<--- LATCHED MSRC 3 H
             B CLK L --->!03                   46!---> MA SELECT S1 H
           PHASE 1 H --->!04                   45!<--- LATCHED MSRC 0 H
       MMUX SEL S1 H <---!05                   44!<--- LATCHED MSRC 4 H
 ENABLE ACV STALL H <---!06                   43!---> MA SELECT S0 H
        MSEQ INIT L --->!07                   42!<--- IRD1 H
             STALL L <---!08                   41!<--- I SIZE 0 L
          LATCH MA L <---!09                   40!<--- LD OSR L
          PREFETCH L <---!10 ....... 39!<--- I SIZE 1 L
       ENA VA SAVE L <---!11 .       . 38!---- GROUND
                 VGA ----!12 . LID . 37!<--- XB PC 00 H


                 VCC ----!13 . DOWN. 36!<--- DST RMODE H
             UTRAP L --->!14 .       . 35!---- GROUND
     M CLK ENABLE H --->!15 ....... 34!<--- LATCHED BUS 0 H
     D CLK ENABLE H --->!16                   33!<--- LATCHED BUS 1 H
     LATCHED BUS 3 H --->!17                   32!<--- PSL CM H
      STATUS VALID L --->!18                   31!<--- LATCHED BUS 2 H
   LATCHED WCTRL 1 H --->!19                   30!<--- XB PC 01 H
        XB1 IN USE L <---!20                   29!<--- BUS 4 H
        XB0 IN USE L <---!21                   28!<--- LATCHED WCTRL 0 H
   LATCHED WCTRL 3 H --->!22                   27!<--- LATCHED WCTRL 2 H
         XB SELECT H <---!23                   26!<--- LATCHED WCTRL 5 H
            ENA PC L <---!24                   25!<--- LATCHED WCTRL 4 H
                                         ---------------
```

PRK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.22 SERVICE AND ARBITRATION CONTROL (SAC-DC617)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for three areas of the processor. They are the System Clock, Service Arbitration, and the IRD counter.

SAC TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE	PIN #	PIN ID	COMMENTS	GATE TYPE
1	CON HALT L		GA1TNF	25	CLK CTRL 1 H		GA1TPF
2	OSC H		GA1TNF	26	CLK CTRL Ø H		GA1TPF
3	CLK DIV3 L		GA1TPF	27	D CLK ENABLE H	I_{OL}=2ØMA	GA1TTN
4	SET PHASE H		GA1TTN	28	M CLK ENABLE H	I_{OL}=2ØMA	GA1TTN
5	MEM STALL H		GA1TNF	29	GEN DEST INH L		GA1TNF
6	SET CLK H		GA1TTN	3Ø	LATCH UTRAP L		GA1TTN
7	BASE CLOCK H		GA1TNF	31	QD CLK ENABLE H		GA1TTN
8	MICRO ADDR INH L		GA1TPF	32	DOUBLE ENABLE H		GA1TNF
9	CS PARITY ERROR H		GA1TNF	33	ENABLE UVECT H		GA1TTN
1Ø	PSL CM H		GA1TNF	34	DO SRVC L		GA1TTF
11	HALT L		GA1TTN	36	LD OSR L		GA1TNF
14	MICFO TRAP L		GA1TNF	37	PSL TP H		GA1TNF
15	IRD CTR 1 H		GA1TTN	39	INT PEND L		GA1TNF
16	BUT Ø H		GA1TNF	4Ø	CS ADDR Ø2 L	I_{OL}=12MA	GA1TCN
17	BUT 2 H		GA1TNF	41	CS ADDR ØØ L	I_{OL}=12MA	GA1TCN
18	BUT 1 H		GA1TNF	42	CS ADDR Ø1 L	I_{OL}=12MA	GA1TCN
19	BUT CONTROL CODE H		GA1TNF	43	TIMER SERVICE H		GA1TNF
2Ø	M CLK L		GA1TNF	44	FPA WAIT L		GA1TNF
21	IRD CTR Ø H		GA1TTN	45	FPA STALL L		GA1TPF
22	IRD CTR 2 H		GA1TTN	46	ARITH TRAP L		GA1TNF
23	MSEQ INIT L		GA1TNF	47	FP TRAP L		GA1TPF
24	INSTR FETCH H		GA1TTN	48	CLKX H		GA1TNF

2. Performance:

2.1 State Elements:

2.1.1 **CLKX FLIP FLOP:** This is a D type flip flop which is clocked with the falling edge of BASE CLOCK H and is set if CLKX H is asserted, the PHASE flipflop is going to be set, and MSEQ INIT L is not asserted. This flop is clocked to zero as long as MSEQ INIT L is asserted.

2.1.2 **RET INH FLIP FLOP:** This is a D type flip flop which is clocked with the rising edge of M CLK L and is set if BUT CONTROL CODE H is asserted, BUT <2:0> = 3, the UTRAP flip flop is clear, and MSEQ INIT L is not asserted. The flop is clocked to zero when MSEQ INIT L is asserted.

2.1.3 **CM IRD1 FLIP FLOP:** This is a D type flip flop which is clocked by the falling edge of BASE CLOCK H. It is set if

PSL CM H is asserted,
and BUT CONTROL CODE H is asserted
and BUT <2:0> = 4 or 5,
and the PHASE flipflop is cleared,
and PHASE is going to be set,
and MSEQ INIT L is not asserted.

Othe'wise it is cleared.

2.1.4 **LD OSR BUF FLIP FLOP:** This is a D type flip flop which is clocked with the falling edge of BASE CLOCF H. The input to the flop is LD OSR L. That is, it is set when LD OSR L is not asserted and vice versa. MSEQ INIT L direct sets the flip flop.

2.1.5 **UTRAP FLIP FLOP:** This is a D type flip flop which is clocked with the falling edge of BASE CLOCK H. It is set if MICRO TRAP L is asserted and MSEQ INIT L is not asserted. Otherwise it is cleared. The flip flop is direct cleared when MICRO TRAP L is not asserted.

2.1.6 **UTRAP BUF FLIP FLOP:** This is a D type flip flop which is clocked with the falling edge of BASE CLOCK H. It is set if MICRO TRAP L is asserted, the UTRAP flip flop is set, and MSEQ INIT L is not asserted. Otherwise it is cleared. The flip flop is direct cleared when MICRO TRAP L is not asserted.

2.1.7 **DELAY CLK CTRL REGISTER:** This is a 2 bit register built from D type flip flops. They are clocked with the falling edge of BASE CLOCK H. The input data to the register is CLK CTRL <1:0> H. Both bits of the register are direct set by MSEQ INIT L.

2.1.8 **CS ERROR FLIP FLOP:** This is a J-K flip flop which is clocked with the rising edge of M CLK L. If MSEQ INIT L is asserted the flip flop is cleared regardless of its state. When the flop is cleared, it can be set if CS PARITY ERROR H is asserted and MSEQ INIT L is not asserted. When the flop is set, it can be cleared if BUT CONTROL CODE H is asserted and BUT <2:0> = 4 or 5, or if MSEQ INIT L is asserted.

2.1.9 **BUT SRVC FLIP FLOP:** This is a D type flip flop which is clocked with the rising edge of M CLK L. When MSEQ INIT L is asserted the flip flop is clocked to Zero for as long as MSEQ INIT is valid. The flip flop is set when INSTR FETCH H is asserted and MSEQ INIT L is not asserted.

2.1.10 **BASIC CLOCK REGISTER:** This is a 2 bit counter built from D type flipflops. They are clocked with the falling edge of OSC H. The count sequence is as follows:

BASIC CLOCK <1>	BASIC CLOCK <0>
0	0
0	1
1	0
0	0....

When CLR DIV3 L is asserted the register is clocked to a zero value.

2.1.11 **CLK ENABLE FLIP FLOP:** This is a D type flip flop which is clocked with the rising edge of OSC H. It is set if BASIC CLOCK <1:0> = 0. The flip flop is direct set when CLR DIV3 L is asserted.

2.1.12 **PHASE FLIP FLOP:** This is a J-K flip flop which is clocked with the falling edge of BASE CLOCK H. BASE CLOCK H changes the state of PHASE if HALT L is not asserted and Internal STOP H is not asserted (see Para. 2.2.6). PHASE is direct cleared when CLR DIV3 H is asserted.

2.2 Signal Descriptions: Certain input pins are defined as having "Skew Latches". These are feed through latches that are open when the clock is not asserted and closed when the clock is asserted.

2.2.1 **OSC H:** This signal is the basic oscillator which is used to generate the System Clocks.

2.2.2 SET CLK H: This is the output of the BASIC CLOCK <1> flip flop.

2.2.3 **BASE CLOCK H:** This clock signal is used to clock a number of state elements and skew latches.

2.2.4 **CLR DIV3 L:** The signal is used only for chip testing purposes. It is used to clear the BASIC CLOCK REGISTER, PHASE, and associated circuitry.

2.2.5 **SET PHASE H:** This signal is the logical OR of two SET-RESET flip flops. If either flip flop is set, SET PHASE H is asserted.

The PSET flip flop indicates when the PHASE flip flop is going to be set. PSET is set if:

BASE CLOCK H is not asserted

and CLR DIV3 L is not asserted
and the PCLR flip flop (described below) is set.

The flip flop is cleared when:

BASE CLOCK H is asserted

or CLR DIV3 L is asserted
or the PCLR flip flop is cleared and the present state of PSET is cleared.

The PCLR flip flop indicates when the PHASE flip flop is going to be cleared. PCLR is set if:

the PHASE flip flop is cleared

and Internal STOP M is not asserted (see Para. 3.3.2.6)
and HALT L is not asserted
and BASE CLOCK H is asserted or the PSET flip flop is set
and CLR DIV3 L is not asserted

or
the PHASE flip flop is set
and Internal STOP M is asserted or HALT L is asserted
and CLR DIV3 L is not asserted
and BASE CLOCK H is asserted or the PSET flip flop is set.

The flip flop is cleared when:

CLR DIV3 L is asserted
or the PSET flip flop is cleared and BASE CLOCK H is not asserted and the present state of PCLR is cleared
or the PHASE flip flop is cleared and (Internal STOP M is asserted or HALT L is asserted).
or the PHASE flip flop is set and Internal STOP M is not asserted and HALT L is not asserted.

2.2.6 M CLK ENABLE H: This signal is the logical OR of two SET-RESET flip flops. If either flip flop is set, M CLK ENABLE H is asserted.

The M1 flip flop is set if:

the CLK ENABLE flip flop is set

and the PCLR flip flop (see Para. 2.2.5) is cleared
and the PHASE flip flop is set

The flip flop is cleared when:

the CLK ENABLE flip flop is cleared
or the PCLR flip flop is set
or the PHASE flip flop is cleared and the present state of
M1 is cleared.

The M2 flip flop is set if:

Internal STOP M is not asserted (See SAC Table 2)

and HALT L is not asserted
and the PHASE flip flop is set
and the CLK ENABLE flip flop is cleared.

The flip flop is cleared when:

Internal STOP M is asserted

or HALT L is asserted
or the PHASE flip flop is cleared
or the CLK ENABLE flip flop is set and the present state of M2 is cleared.

2.2.7 D CLK ENABLE H: This signal is the logical OR of two SET-RESET flip flops. If either flip flop is set, D CLK ENABLE H is asserted.

The D1 flip flop is set if:

the CLK ENABLE flip flop is set
and the PCLR flip flop (see Para. 2.2.5) is cleared
and the PHASE flip flop is set and Internal STOP D is not asserted (See SAC Table 3).

The flip flop is cleared when:

the CLK ENABLE flip flop is cleared

or the PCLR flip flop is set
or the present state of D1 is cleared and (the PHASE flip flop is cleared OR Internal STOP D is asserted).
The D2 flip flop is set if:

Internal STOP M is not asserted

and Internal STOP D is not asserted
and the PHASE flip flop is set
and HALT L is not asserted
and the CLK ENABLE flip flop is cleared.

The flip flop is cleared when:

Internal STOP M is asserted

or Internal STOP D is asserted
or the PHASE flip flop is cleared
or HALT L is asserted
or the CLK ENABLE flip flop is set and the present state of D2 is cleared.

2.2.8 QD CLK ENABLE H: This signal is asserted as follows:

DOUBLE ENABLE H	HALT L	INTERNAL STOP M	INTERNAL STOP D	PHASE FLIP FLOP	QD CLK ENABLE H
X	L	X	X	X	L
H	H	X	X	X	H
L	H	NOT ASSERTED	NOT ASSERTED	SET	H
OTHERWISE					L

SAC TABLE 2

MSEQ INIT L	MEM STALL H	FPA WAIT L	FPA STALL L	CLKX FLIP FLOP	CM IRD1 FLIP FLOP	MICRO TRAP L	UTRAP BUF FLIP FLOP	LD OSR L	LD OSR BUF FLIP FLOP	PHASE FLIP FLOP	INTERNAL STOP M
L	X	X	X	X	X	X	X	X	X	X	NOT ASSERTED
H	H	X	X	X	X	X	X	X	X	X	ASSERTED
H	X	X	X	SET	X	X	X	X	X	X	ASSERTED
H	X	X	X	X	SET	X	X	X	X	X	ASSERTED
H	X	X	X	X	X	L	CLEAR	X	X	X	ASSERTED
H	X	X	X	X	X	X	X	H	CLEAR	SET	ASSERTED
H	X	L	L	X	X	X	X	X	X	X	ASSERTED
		OTHERWISE									NOT ASSERTED

SAC TABLE 3

GEN DEST INH L	MSEQ INIT L	RE INH FLIP FLOP	INTERNAL STOP D
L	X	X	ASSERTED
X	L	X	ASSERTED
X	X	L	ASSERTED
H	H	H	NOT ASSERTED

2.2.9 **MEM STALL H:** This signal is used to generate:

M CLK ENABLE H
D CLK ENABLE H
QD CLK ENABLE H
SET PHASE H
HALT L

2.2.10 **FPA WAIT L:** This signal in conjunction with FPA STALL L is used to form:

M CLK ENABLE H
D CLK ENABLE H
QD CLK ENABLE H
SET PHASE H
HALT L

2.2.11 **FPA STALL L:** This signal comes from the FPA. For its use see Para. 2.2.10.

The pullup for this Open Collector signal is included here such that it will be unasserted if nothing is connected externally.

2.2.12 **CLKX H:** This signal specifies that the processor should stall for one bus cycle while the PHASE flipflop is set. It is used as the direct input to the CLKX flip flop which is included in the generation of:

M CLK ENABLE H
D CLK ENABLE H
QD CLK ENABLE H
SET PHASE H
HALT L

2.2.13 **GEN DEST INH L:** This signal is used to create:

D CLK ENABLE H
QD CLK ENABLE H

2.2.14 **DOUBLE ENABLE H:** This signal is used to generate QD CLK ENABLE H.

2.2.15 **HALT L:** This signal is used to inhibit all system clocks for debugging. It is generated by a J-K flip flop which changes on the falling edge of BASE CLOCK H. The flop is direct cleared by MSEQ INIT L. The setting and clearing of the HALT flop is controlled by CLK CTRL <1:0> H and the DELAY CLK CTRL register. HALT is also set if two CS PARITY ERRORs occur and the second occurs before the micro-code has the chance to save information about the first one.

The encoding of the CLK CTRL signals is as follows:

CLK CTRL <1:0> H	FUNCTION
3	RUN
2	SINGLE MICRO-INSTRUCTION
1	SINGLE TICK
0	HALT

The legal transitions of functions are:

RUN --> HALT (Must occur in 2 bus cycles)
HALT --> Single Micro-Instruction --> HALT
HALT --> Single Tick --> HALT
HALT --> RUN

See SAC Table 4.

2.2.16 **CLK CTRL <1:0> H:** The pullups for these Open Collector signals are included here such that they will be asserted if nothing is connected externally. They are received through skew latches clocked with BASE CLOCK H and it is the latched versions that are actually used throughout this specification. See Para. 2.2.15 for functional description.

2.2.17 **DO SRVC L:** This signal indicates that a service request is present and control should be transferred to the appropriate routine. DO SRVC L is generated as follows. (See SAC Table 5).

2.2.18 **CS ADDR <2:0> L:** These are the address lines used to access the Control Store ROMs. If DO SRVC L is valid then these signals specify the address of the service routine. (See SAC Table 6).

2.2.19 **ARITH TRAP L:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H.

SAC TABLE 4

CLK	DELAY			CS	CS					
CTRL	CLK			PARITY		ERROR	MSEQ	HALT	BASE	
<1:Ø>	CTRL	PHASE	INTERNAL	ERROR		FLIP	INIT	FLIP	CLOCK	HALT
H	<1:Ø>	H	STOP H	H	FLOP		L	FLOP	H	L
X	X	X	X	X	X	L	CLEAR	X	H	
X	X	L	UNASSERTED	H	SET	H	CLEAR	-> SET	H -> L	H -> L
X	X	H	ASSERTED	H	SET	H	X	H -> L	X -> L	
X	X	H	X	H	SET	H	SET	H -> L	L	
3	X	X	X	L	OR	CLEAR	H	SET -> CLEAR	H -> L	L -> H
2	Ø	X	X	L	OR	CLEAR	H	SET -> CLEAR	H -> L	L -> H
1	Ø	X	X	L	OR	CLEAR	H	SET -> CLEAR	H -> L	L -> H
1	Ø,1,2	X	X	L	OR	CLEAR	H	CLEAR -> SET	H -> L	H -> L
2	X	L	UNASSERTED	L	OR	CLEAR	H	CLEAR -> SET	H -> L	H -> L
Ø	X	L	UNASSERTED	L	OR	CLEAR	H	CLEAR -> SET	H -> L	H -> L
	OTHERWISE							NO CHANGE	H -> L	NO CHANGE

SAC TABLE 5

	CS							
BUT	PARITY	ARITH	FP	TIMER	CON	INT	PSL	DO
SRVC	ERROR	TRAP	TRAP	SERVICE	HALT	PEND	TP	SRVC
FLOP	H	L	L	H	L	L	H	L
CLEAR	X	X	X	X	X	X	X	H
X	H	X	X	X	X	X	X	H
SET	L	L	X	X	X	X	X	L
SET	L	X	L	X	X	X	X	L
SET	L	X	X	H	X	X	X	L
SET	L	X	X	X	L	X	X	L
SET	L	X	X	X	X	L	X	L
SET	L	X	X	X	X	X	H	L
	OTHERWISE							H

SAC TABLE 6

	CS	MICRO							
BUT	PARITY	ADDR	ARITH	FP	TIMER	CON	INT	PSL	CS
SRVC	ERROR	INH	TRAP	TRAP	SERVICE	HALT	PEN	TP	ADDR
FLOP	H	L	L	L	H	L	L	H	<2:0>
CLEAR	X	X	X	X	X	X	X	X	0
X	H	X	X	X	X	X	X	X	0
X	X	L	X	X	X	X	X	X	0
SET	L	H	L	X	X	X	X	X	1
SET	L	H	H	L	X	X	X	X	2
SET	L	H	H	H	H	X	X	X	4
SET	L	H	H	H	L	L	X	X	6
SET	L	H	H	H	L	H	L	X	0
SET	L	H	H	H	L	H	H	H	5

2.2.20 **FP TRAP L:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H. The pullup for this Open Collector signal is included here such that it will not be asserted if nothing is connected externally.

2.2.21 **TIMER SERVICE H:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H.

2.2.22 **COM HALT L:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H.

2.2.23 **INT PEND L:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H.

2.2.24 **PSL TP H:** This signal is used to form DO SRVC L, CS ADDR<2:0> and ENABLE UVECT H.

2.2.25 **ENABLE UVECT H:** This signal is used to enable the Memory Interface micro-vector onto the CS ADDR lines. It is generated as follows:

PHASE FLIP FLOP	MSEQ INIT L	DO SRVC L	UTRAP FLIP FLOP	ARITH TRAP L	FP TRAP L	TIMER SERVICE H	CON HALT L	INT PEND L	PSL TP H	ENABLE UVECT H
CLEAR	X	X	X	X	X	X	X	X	X	L
X	L	X	X	X	X	X	X	X	X	L
SET	H	H	CLEAR	X	X	X	X	X	X	L
SET	H	H	SET	X	X	X	X	X	X	H
SET	H	L	X	L	X	X	X	X	X	L
SET	H	L	X	H	L	X	X	X	X	L
SET	H	L	X	H	H	H	X	X	X	L
SET	H	L	X	H	H	L	L	X	X	L
SET	H	L	X	H	H	L	H	L	X	H
SET	H	L	X	H	H	L	H	H	H	L
Otherwise										L

2.2.26 INSTR FETCH H: This signal indicates that an IRD1 branch has been attempted. INSTR FETCH H is generated as follows:

BUT CONTROL CODE H	BUT <2:0> H	GEN DEST INH L	INSTR FETCH H
L	X	X	L
H	0,1,2,3,6,7	X	L
H	4,5	H	H
H	4,5	L	L
OTHERWISE			L

2.2.27 CS PARITY ERROR H: This signal is used to generate HALT L, DO SRVC L, and CS ADDR <2:0> L.

2.2.28 MICRO TRAP L: This signal is used to generate

ENABLE UVECT H
INSTR FETCH H
IRD CTR<2:0> H

2.2.29 LATCH UTRAP L: This is a latched version of MICRO TRAP L.

The signal is asserted if:

the PHASE flip flop is set

and the UTRAP flip flop is set.

2.2.30 MICRO ADDR INH L: This signal can be asserted by various external devices. Its purpose is to disable the CS ADDR lines. The pullup for this signal is included here such that it will not be asserted if nothing is connected externally.

2.2.31 M CLK L: This is the clock associated with the micro-sequencer. The rising edge of M CLK is used to clock the:

CS ERROR flip flop
RET INH flip flop
BUT SRVC flip flop
and the IRD CTR.

2.2.32 MSEQ INIT L: This is an initialization signal which is used to clear most of the state in this chip. When it is asserted, it

direct sets the LD OSR BUF flip flop
forces CLKX flop to be clocked to ZERO
D CLK ENABLE H is unasserted
QD CLK ENABLE H is unasserted
forces RET INH flip flop to be clocked to ZERO
direct clears the HALT flip flop
HALT L is not asserted
Previous CLK CTRL <1:Ø> H are forced to the RUN function
forces the BUT SRVC flip flop to be clocked to ZERO
direct clears the IRD CTR.
forces the CM IRD1 flip flop, the UTRAP BUF flip flop, the UTRAP BUF flip flop, and the CS ERROR flip flop to be clocked to ZERO

2.2.33 IRD CTR <2:Ø> H: This is a 3 bit counter which keeps track of the operands of a macro-instruction. It is direct cleared by MSEQ INIT L. The rising edge of M CLK L clocks the counter which is a D type register. The counter changes as follows (See SAC Table 6).

2.2.34 BUT CONTROL CODE H: This signal indicates that BUT <5:3> = Ø and is used to generate

D CLK ENABLE H
QD CLK ENABLE H
DO SRVC L
CS ADDR <2:Ø>
INSTR FETCH H
and to control the RET INH flop
BUT SRVC flop
and the IRD CTR.

It is received through a skew latch clocked with M CLK L and it is the latched version which is used throughout this specification.

2.2.35 BUT <2:Ø> H: These signals come from the Control Store latches. For their use see Para. 2.2.34. They are received through skew latches clocked with M CLK L and it is the latched versions which are actually used throughout this specification.

2.2.36 **PSL CM H:** This signal indicates that the machine is running Compatibility Mode Instructions. It is used to generate:

IRD CTR <2:0> H
and the CM IRD1 flip flop.

2.2.37 LD OSR L: This signal is used to generate IRD CTR <2:0> H and M CLK ENABLE H. It is received througn a skew latch clocked with BASE CLOCK H and it is the latched version which is used throughout this specification.

SAC TABLE 7

				PRESENT			NEXT	
		BUT		VALUE			VALUE	IRD
LD	PSL	CONT.	BUT	IRD	DO	UTRAP	IRD	CTR
OSR	CM	CODE	<2:0>	CTR	SRVC	FLIP	CTR	<2:0>
L	H	H	H	REG.	L	FLOP	REGISTER	H
X	X	H	4,5	X	H	CLEAR	0	7
X	X	X	X	X	L	X	PRESENT VALUE	IRD CTR REG
X	X	X	X	X	X	SET	PRESENT VALUE	IRD CTR REG
L	X	L	X	X	H	CLEAR	PRESENT VALUE +1	IRD CTR REG
L	X	H	0-3,6	X	H	CLEAR	PRESENT VALUE +1	IRD CTR REG
L	X	H	7	X	H	CLEAR	PRESENT VALUE	IRD CTR REG
H	X	L	X	X	H	CLEAR	PRESENT VALUE	IRD CTR REG
H	L	H	0-3,6,7	X	H	CLEAR	PRESENT VALUE	IRD CTR REG
H	H	H	1	0,1	H	CLEAR	FRESENT VALUE +1	IRC CTR RFG
		OTHERWISE					PRESENT VALUE	IRD CTR REG

```
                                 ---------------
           CON HALT L --->!01                  48!<--- CLKX H
                OSC H --->!02                  47!<--- FP TRAP L
           CLR DIV3 L --->!03                  46!<--- ARITH TRAP L
          SET PHASE H <---!04                  45!<--- FPA STALL L
          MEM STALL H --->!05                  44!<--- FPA WAIT L
            SET CLK H <---!06                  43!<--- TIMER SERVICE H
         BASE CLOCK H --->!07                  42!---> CS ADDR 01 L
     MICRO ADDR INH L --->!08                  41!---> CS ADDR 00 L
    CS PARITY ERROR H --->!09                  40!---> CS ADDR 02 L
             PSL CM H --->!10 ........ 39!<--- INT PEND L
               HALT L <---!11 .      . 38!---- GROUND
                  VGA ----!12 . LID . 37!<--- PSL TP H
                  VCC ----!13 . DOWN. 36!<--- LD OSR L
         MICRO TRAP L --->!14 .      . 35!---- GROUND
          IRD CTR 1 H <---!15 ........ 34!---> DO SRVC L
              BUT 0 H --->!16                  33!---> ENABLE UVECT H
              BUT 2 H --->!17                  32!<--- DOUBLE ENABLE H
              BUT 1 H --->!18                  31!---> QD CLK ENABLE H
 BUT CONTROL CODE H --->!19                  30!---> LATCH UTRAP L
              M CLK L --->!20                  29!<--- GEN DEST INH L
          IRD CTR 0 H <---!21                  28!---> M CLK ENABLE H
          IRD CTR 2 H <---!22                  27!---> D CLK ENABLE H
          MSEQ INIT L --->!23                  26!<--- CLK CTRL 0 H
        INSTR FETCH H <---!24                  25!<--- CLK CTRL 1 H
                                 ---------------
```

SAC FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.23 SCRATCH PAD ADDRESS CHIP (SPA-DC616)

1. GENERAL DESCRIPTION:

The SPA chip serves two main purposes:

(A) To control the operation of two external arrays of scratch pad, RAM-R and RAM-M. This is done by controlling their scratch pad address and chip select signals.

(B) To provide a mechanism to undo the auto-incrementing and the auto-decrementing of the general purpose registers. This is done via the register back-up stack (RBS).

SPA TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	IRD RNUM 1 H		GA1TNF
2	D SIZE 0 H		GA1TNG
3	IRD LD RNUM H		GA1TNF
4	D CLK ENABLE H		GA1TNH
5	MSRC 3 H		GA1TNF
6	M CLK L		GA1TNF
7	DP PHASE H		GA1TNH
8	MSRC 2 H		GA1TNF
9	MSRC 1 H		GA1TNF
10	MSRC 0 H		GA1TNF
11	MSRC 4 H		GA1TNF
14	RSPA 1 H		GA1TTG
15	MSPA 1 H		GA1TTG
16	MSPA 2 H		GA1TTG
17	RSPA 2 H		GA1TTG
18	MTMP CS L		GA1TTN
19	RSPA 0 H		GA1TTG
20	MSPA 0 H		GA1TTG
21	RSPA 3 H		GA1TTG
22	MSPA 3 H		GA1TTG
23	RSRC 2 H		GA1TNF
24	SPW M L		GA1TNF
25	RCS TMP L		GA1TTN
26	RCS IPR L		GA1TTN
27	RSRC 0 H		GA1TNF
28	LIT 0 H		GA1TNH
29	RSRC 5 H		GA1TNF
30	RSRC 4 H		GA1TNF
31	DST RMODE H		GA1TNF
32	RSRC 3 H		GA1TNF
33	LITRL EN L		GA1TTN
34	RSRC 1 H		GA1TNF
36	RCS GPR L		GA1TTN
37	INSTR FETCH H		GA1TNF
39	D SIZE 1 H		GA1TNG
40	SPASTA 1 H		GA1TTN
41	SPASTA 0 H		GA1TTN
42	WBUS 01 H		GA1TZG
43	WBUS 00 H		GA1TZG
44	IRD RNUM 0 H		GA1TNF
45	WBUS 03 H		GA1TZG
46	WBUS 02 H		GA1TZG
47	IRD RNUM 2 H		GA1TNF
48	IRD RNUM 3 H		GA1TNF

2. Performance:

2.1 Notations:

Unless otherwise specified, the notations used in this document are as follows:

(A) all numbers are in decimal;

(B) signal values are in voltage level:
H = High
L = Low
X = Don't care (Hign or Low)

2.2 **General Information:** The scratch pad RAM chips to be controlled by the SPA chip are 16 word by 4 bit RAMs.

2.2.1 **RAM-M:** RAM-M contains 16 locations used for microcode temporaries. Data from RAM-M are output to the MBUS of the CPU data path. The scratch pad location is specified by the MSRC field of the microword. The assignment and some useful notations for the MSRC field are defined in SPA Table 2.

SPA TABLE 2

MSRC FIELD ASSIGNMENT

MSRC(hex)	Assignment	Denoted by
00-0F	MTMP, 0-15	MTMP
10	MTMP[RNUM]	MTMP
11	MTMP[RNUM+1]	MTMP
12	Reserved	Reserved
13	Reserved	Reserved
14	PUSH-	PUSH-
15	PUSH+	PUSH+
16	WBUS<-RNUM	WBUS<-RNUM
17	Reserved	Reserved
18	Reserved	Reserved
19	Reserved	Reserved
1A	Reserved	Reserved
1B	Reserved	Reserved
1C	READ RBS	READ RBS
1D	RNUM<-WBUS	RNUM<-RBS
1E	WBUS<-RBSP	WBUS<-RBSP
1F	Reserved	Reserved

note: Reserved assignments are for memory interface.

2.2.2 /RAM-R: RAM-R contains 48 scratch pad locations. The 48 locations are intended to be used for the following purposes:

Location	Purpose
0 - 15	Microcode Temporaries, RTMP0-RTMP15.
16 - 30	General Purpose Registers, GPR0-GPR15.
31 - 47	Internal Processor Registers, IPR0-IPR15.

Data from RAM-R are output to the RBUS of the CPU data path. The scratch pad location is specified by the RSRC field of the microword. The assignment and some useful notations for the RSRC field are defined in SPA Table 3.

RSRC (hex)	SPA Table 3 RSRC Field Assignment	Denoted by
00-0F	RTMP, 0-15	RTMP
10-1F	GPR, 0-15	GPR
20-2F	IPR, 0-15	IPR
30	RTMP[RNUM]	RTMP
31	DST[RNUM]	DST
32	IPR[RNUM]	IPR
33	GPR[RNUM]	GPR
34	Reserved	RSRC.DEF
35	LONG LIT	RSRC.DEF
36	ZERO	RSRC.DEF
37	CLR RBSP	RSRC.DEF
38	RTMP[RNUM.or.1]	RTMP
39	DST[RNUM.or.1]	DST
3A	IPR[RNUM.or.1]	IPR
3B	GPR[RNUM.or.1]	GPR
3C	RTMP[RNUM+1]	RTMP
3D	DST[RNUM+1]	DST
3E	IPR[RNUM+1]	IPR
3F	GPR[RNUM+1]	GPR

NOTE: RSRC assignment and the definitions of some useful notations.

2.2.3 Scratch pad write and clock signals: To provide some flexibility to the microcode in using the temporaries, RAM-R and RAM-M are implemented so that for i=0 through 7, writing to RTMP<i> always implys a simultaneous write to MTMP<i> and vice versa. Implementation of this feature imposes that during the first half of the cycle, the SPA generates the scratch pad address/chip selects for a read operation. During the second half of the cycle, the SPA generates a potentially different set of address/chip selects for a write operation.

Clock signals used by the SPA chip are the M CLK L and the DP PHASE H. Both clock signals are used to define the relative time frame of a microcycle.

The time period starting from the rising edge of M CLK L and terminated with the rising edge of DP PHASE H is defined as the "first half of the cycle". The time period starting from the rising edge of DP PHASE H and terminated with the rising edge of M CLK L is defined as the "second half of the cycle".

2.3 Pin Functions:

2.3.1 **LIT 0 H:** An input from the microword which, when High, indicates that the RSRC field is used to form a literal. Under this condition, the SPA chip will default the RSRC to be the same as LLLHHH, i.e. addressing RTMP7 directly.

2.3.2 **MSRC <4:0> H:** A five bit field from the microword to control RAM-M mainly. See SPA Table 2 for its assignment.

2.3.3 **RSRC <5:0> H:** A six bit field from the microword used to control RAM-R mainly. See SPA Table 3 for its assignment.

2.3.4 **SPW M L:** A decoded signal from the SPW field of the microword.

SPW M L	Meaning
L	Enable write to RAM-M
H	Enable write to RAM-R

This signal is used to control the scratch pad address and the chip select outputs during the second half of the cycle. See 3.4.3 and 3.4.4.

2.3.5 **M CLK L:** A clock signal used to load all memory elements within the chip, generally this is done at the rising edge of the clock.

2.3.6 **D CLK ENABLE H:** An input signal used to enable the loading of RNUM and RBSP. See sections 2.4.1 and 2.4.2.2.

2.3.7 **DP PHASE H:** A clock signal used primarily to control the scratch pad address and the chip select signals. It is also used to generate an internal clock for the RBS, see 2.4.2.1.

2.3.8 **DST RMODE H:** An input signal used to indicate whether the destination mode of the current macro instruction is Register mode or not.

DST RMODE H	Meaning
H	Register mode
L	non register mode

The signal is used to conditionally disable the writing to the GPR scratch pad locations when the RSRC=DST and that DST RMODE H is not asserted. See SPA Table 8.

2.3.9 IRD RNUM <3:0> H: This is one of the three sources to be loaded to RNUM. The four signals are usually the register number field of an operand specifier.

2.3.10 IRD LD RNUM H: This is a control signal used to enable loading the IRD RNUM <3:0> H to RNUM. See section 2.4.1.

2.3.11 D SIZE <1:0> H: The two signals used to indicate the data size. When the MSRC=PUSH+ or PUSH-, the two signals are loaded into RBS as described in section 2.4.2.1.

2.3.12 INSTR FETCH H: An input signal to indicate that the micro machine is in the state of fetching a new macro instruction. When asserted, the RBSP will be cleared at M CLK L = L. See section 2.4.2.2.

2.3.13 WBUS <3:0> H: These are four tri-state pins used to interface the SPA chip with the WBUS of the CPU data path.

Output to the WBUS is controlled by the MSRC alone:

MSRC	Effect on WBUS
WBUS <- RNUM	WBUS <3:0> H = RNUM
WBUS <- RBSP	WBUS <3:0> H = 0'RBSP
READ RBS	WBUS <3:0> H = f(RBS)
All others	output disabled

NOTE: f(RBS) IS DEFINED AS

RBS<5:4>	f(RBS)
00	1
01	2
10	4
11	8

2.3.14 MSPA <3:0> H: The four scratch pad address signals output to RAM-M. See section 2.4.3.1 for definitions.

2.3.15 RSPA <3:0> H: The four scratch pad address signals output to RAM-R. See section 2.4.4.1 for definitions.

2.3.16 **MTMP CS L:** The chip select signal output to RAM-M. See section 2.4.3.2.

2.3.17 **RCS TMP L, RCS GPR L, RCS IPR L:** The chip select signal output to RAM-R. See section 2.4.4.2.

2.3.18 **LITRL EN L:** A deccded signal from RSRC <5:0> H and LIT 0 H. When asserted, the signal would enable an external register (used to hold a microcode literal) to source data onto the RBUS.

The signal is defined as follows:

```
if (first half of the cycle) .and. (LIT 0 H = L) .and.
(RSRC = LONG LIT)
          then LITRL EN L = L else LITRL EN L = H.
```

2.3.19 **SPASTA <1:0> H:** The two status output from the SPA chip used for micro-branches. See section 2.4.5 for definitions.

2.4 Block Diagram:

Figure 3 depicts the block diagram of the SPA chip. There are five major sections.

A. RNUM register
B. RBS, Register Back up Stack
C. Address and chip select for RAM-M
D. Address and chip selects for RAM-R
E. STATUS

2.4.1 **RNUM (Register NUMber Register):** The RNUM is a 4 bit D-type register used to indirectly address a scratch pad location.

There are three sources of data to load RNUM:

A. A 4 bit number from the microsequencer (IRD RNUM <3:0> H)

B. WBUS <3:0> H

C. The register back up stack(RBS).

IRD LD	D EN	MSRC	Data loaded
H	X	X	IRD RNUM <3:0> H
L	H	RNUM <- WBUS	WBUS <3:0> H
L	H	READ RBS	RBS<3:0>, (see 2.4.2)

where: IRD LD=IRD LD RNUM H
D EN =D CLK ENABLE H

When the above condition is satisfied, RNUM is loaded at the rising edge of the M CLK L.

2.4.2 **Register Back-Up Stack (RBS):** The RBS is a 6 word by 7 bit stack, pointed to by a three bit register called the RBSP. (RBS<i:j> will be used to denote the bit position <i:j> of the stack location pointed to by RBSP.)

2.4.2.1 **The Basic Stack:** The stack itself is made up of six transparent latches of seven bits wide each. When the MSRC specifies a PUSH- or a PUSH+ operation, data is loaded into the stack location pointed to by the RBSP. The latching occurs during time interval t, where t is defined ~~as~~ the time period starting from the rising edge of DP PHASE H and terminated with the falling edge of M CLK L.
The data loaded are the RAM-R scratch pad address, the D SIZE <1:0> H, and whether the MSRC specifies a PUSH- or a PUSH+ operation:

```
RBS<3:0> <- RSPA <3:0> H;
RBS<5:4> <- D SIZE <1:0> H;
RBS<6>   <- if PUSH+ then 1 else 0.
```

2.4.2.2 **The Pointer, RBSP:** The RBSP is a three bit D-type register. It can either be cleared or incremented:

IFCH	MSRC	D EN	LIT	RSRC	RBSP is
H	X	X	X	X	cleared
X	X	X	L	CLR RBSP	cleared
L	PUSH+	H	X	not(CLR RBSP)	incremented
L	PUSH-	H	X	not(CLR RBSP)	incremented
L	READ RBS	H	X	not(CLR RBSP)	incremented
ALL OTHER CONDITIONS - - - - - - - - - - - - - -					NO CHANGE

where: IFCH = INSTR FETCH H
LIT = LIT 0 H
D EN = D CLK ENABLE H

When being cleared, the RBSP is loaded with 0 when the M CLK L = L.

When being incremented, the RBSP is loaded with RBSP + 1 at the rising edge of the M CLK L.

2.4.3 **Address and Chip Select for RAM-M:**

2.4.3.1 **Scratch Pad Address:** The scratch pad address to RAM-M appears on the four MSPA <3:0> H pins. There are four sources for this set of signals:

MSRC <3:0> H, directly from the microword
RNUM, contents of RNUM

kNUM + 1, 1 plus the contents of RNUM
RSPA <3:0> H, the same address for RAM-R

During the first half of the cycle, the MSPA <3:0> H signals are specified by the MSRC alone, as defined in SPA Table 4.

SPA TABLE 4

MSRC AND MSPA DATA

MSRC	**MSPA <3:0> H**
MTMP, 0-15	MSRC <3:0> H
MTMP[RNUM]	RNUM
MTMP[RNUM+1]	RNUM + 1
all others	0

Definition of MSPA <3:0> H during the first half of the cycle.

During the second half of the cycle, the MSPA signals would either remain the same or switch to the address output for RAM-R, as determined by the SPW M L signal.

SPW M L	MSPA <3:0> H
L	Remains the same
H	duplicate RSPA <3:0> H

2.4.3.2 **Chip Select for RAM-M:** One pin is used to enable the chip select for the 16 location RAM-M.
During the first half of the cycle, the chip select is used to read data onto the MBUS. The signal is defined as:

```
if MSRC = Reserved, then MTMP CS L = H
                    else MTMP CS L = L.
```

During the second half of the cycle, the chip select is used to enable a write to RAM-M. SPA Table 5 lists the conditions under which the chip select is asserted.

SPA TABLE 5

CHIP SELECT ASSERTION DATA

SPWM	LIT	RSRC	RNUM	MTMP CS L
L	X	X	X	L
H	H	X	X	L
H	L	RTMP, Ø-7	X	L
H	L	RTMP, 8-15	X	H
H	L	RTMP[RNUM]	Ø-7	L
H	L	RTMP[RNUM]	8-15	H
H	L	Reserved	X	L
H	L	LONG LIT	X	L
H	L	ZERO	X	L
H	L	CLR RBSP	X	L
H	L	RTMP[RNUM.or.1]	Ø-7	L
H	L	RTMP[RNUM.or.1]	8-15	H
H	L	RTMP[RNUM+1]	Ø-6,15	L
H	L	RTMP[RNUM+1]	7-14	H
H	L	All others	X	H

where: SPWM = SPW M L
LIT = LIT Ø H

NOTE: Signal definition for the RAM-M chip select signal during the second half of the cycle. Note that the signal is not affected by MSRC.

2.4.4 Address and Chip Selects for RAM-R:

2.4.4.1 Scratch Pad Address, RSPA: The scratch pad address to RAM-R appears on the four RSPA <3:Ø> H pins. There are five sources for this set of signals:

A. RSRC <3:Ø> H, directly from the microword
B. RNUM, contents of RNUM
C. RNUM + 1, 1 plus the contents of RNUM
D. RNUM.or.1, RNUM<3:1>'1
E. MSPA <3:Ø> H, the same address for RAM-M.

During the first half of the cycle, the RSPA <3:Ø> H signals are specified in SPa Table 6.

SPA TABLE 6

RSPA<3:0>H SIGNALS DATA

LIT	RSRC	RSPA <3:0> H
H	X	7
L	RTMP, 0-15	RSRC <3:0> H
L	GPR, 0-15	RSRC <3:0> H
L	IPR, 0-15	RSRC <3:0> H
L	RTMP[RNUM]	RNUM
L	DST[RNUM]	RNUM
L	IPR[RNUM]	RNUM
L	GPR[RNUM]	RNUM
L	Reserved	0
L	LONG LIT	0
L	ZERO	0
L	CLR RBSP	0
L	RTMP[RNUM.or.1]	RNUM<3:1>'1
L	DST[RNUM.or.1]	RNUM<3:1>'1
L	IPR[RNUM.or.1]	RNUM<3:1>'1
L	GPR[RNUM.or.1]	RNUM<3:1>'1
L	RTMP[RNUM+1]	RNUM + 1
L	DST[RNUM+1]	RNUM + 1
L	IPR[RNUM+1]	RNUM + 1
L	GPR[RNUM+1]	RNUM + 1

where: LIT = LIT 0 H

Definition of the RSPA <3:0> H during the first half of the cycle.

During the second half of the cycle, the RSPA signals would either remain the same or switch to the address output for RAM-M, as determined by the SPW M L signal.

SPW M L	RSPA <3:0> H
H	Remains the same
L	duplicate MSPA <3:0> H

(Note that during the second half of the cycle, either the MSPA or RSPA outputs switch, but never both.)

2.4.4.2 **Chip Selects for RAM-R:** There are three chip select signals for RAM-R, with one for each group of the scratch pad locations:

Chip select	used for
RCS TMP L	RTMPØ-RTMP15
RCS GPR L	GPRØ-GPR15
RCS IPR L	IPRØ-IPR15

During the first half of the cycle, the chip selects are used to read a location from only one group of the scratch pad locations. SPA Table 7 lists the conditions under which a signal is asserted.

SPA TABLE 7

SIGNAL ASSERTION DATA

LIT	RSRC	T	G	I
H	anything	L	H	H
L	RTMP	L	H	H
L	DST	H	L	H
L	IPR	H	H	L
L	GPR	H	L	H
L	RSRC.DEF	H	H	H

where: LIT = LIT Ø H
T = RCS TMP L
G = RCS GPR L
I = RCS IPR L

Definitions of the chip selects for RAM-R
during the first of the cycle.

During the second half of the cycle, the chip selects are used to enable a write to a location in only one group of the scratch pad locations. The chip selects for GPR's and IPR's are specified in SPA Table 8, and the chip select for RTMP is specified in SPA Table 9.

SPA TABLE 8

CHIP SELECT DATA (FOR GPR'S & IPR'S)

SPWM	LIT	RSRC	RMODE	G	I
L	X	X	X	H	H
H	H	X	X	H	H
H	L	RTMP	X	H	H
H	L	DST	H	L	H
H	L	DST	L	H	H
H	L	IPR	X	H	L
H	L	GPR	X	L	H
H	L	RSRC.DEF	X	H	H

where: LIT = LIT 0 H
SPWM = SPW M L
RMODE = DST RMODE H
G = RCS GPR L
I = RCS IPR L

Definitions of the chip selects for the GPR's and the IPR's during the second half of the cycle.

SPA TABLE 9

CHIP SELECT DATA (FOR RTMP)

SPWM	LIT	RSRC	MSRC	RNUM	T
H	H	X	X	X	L
H	L	RTMP	X	X	L
H	L	RSRC.DEF	X	X	L
H	L	GPR	X	X	H
H	L	IPR	X	X	H
H	L	DST	X	X	H
L	X	X	MTMP, 0-7	X	L
L	X	X	MTMP, 8-15	X	H
L	X	X	MTMP[RNUM]	0-7	L
L	X	X	MTMP[RNUM]	8-15	H
L	X	X	MTMP[RNUM+1]	0-6,15	L
L	X	X	MTMP[RNUM+1]	7-14	H
L	X	X	all others	X	L

where: SPWM = SPW M L
LIT = LIT 0 H
T = RCS TMP L

Definition of the RCS RTMP L signal during the second half of the cycle.

2.4.5 **Status:** The STATUS logic generates two status signals, SPASTA <1:0> H, useful for micro-branches. The status output is determined by both the RSRC and the MSRC.

When RSRC =(GPR or DST), signal definitions for STATUS <1:0> H are:

MSRC	SPASTA <1:0> H
READ RBS	Undefined
RNUM<-WBUS	Undefined
WBUS<-RBSP	Undefined
All others	LL : RNUM = any value but 6,7 or 14
	LH : RNUM = 14
	HL : RNUM = 7
	HH : RNUM = 6

When RSRC = anything but (GPR or DST), signal definitions for STATUS <1:0> H are:

MSRC	SPASTA <1:0> H
RNUM<-WBUS	LL : WBUS = 8,9,10,11,12, or 13
	HL : WBUS = 5,6,7,14, or 15
	HH : WBUS = 0,1,2,3, or 4
READ RBS	if RBS<6>=1 then HL else LL
WBUS<-RBSP	if RBSP = 0 then LH else LL
All others	LL

```
                          ---------------
      IRD RNUM 1 H --->!Ø1          48!<--- IRD RNUM 3 H
        D SIZE Ø H --->!Ø2          47!<--- IRD RNUM 2 H
     IRD LD RNUM H --->!Ø3   SPA    46!<--> WBUS Ø2 H
   D CLK ENABLE H --->!Ø4          45!<--> WBUS Ø3 H
          MSRC 3 H --->!Ø5          44!<--- IRD RNUM Ø H
           M CLK L --->!Ø6          43!<--> WBUS ØØ H
        DP PHASE H --->!Ø7          42!<--> WBUS Ø1 H
          MSRC 2 H --->!Ø8          41!---> SPASTA Ø H
          MSRC 1 H --->!Ø9          4Ø!---> SPASTA 1 H
          MSRC Ø H --->!1Ø ....... 39!<--- D SIZE 1 H
          MSRC 4 H --->!11 .     . 38!---- GROUND
               VGA ----!12 . LID . 37!<--- INSTR FETCH H
               VCC ----!13 . DOWN. 36!---> RCS GPR L
          RSPA 1 H <---!14 .     . 35!---- GROUND
          MSPA 1 H <---!15 ....... 34!<--- RSRC 1 H
          MSPA 2 H <---!16          33!---> LITRL EN L
          RSPA 2 H <---!17          32!<--- RSRC 3 H
         MTMP CS L <---!18          31!<--- DST RMODE H
          RSPA Ø H <---!19          3Ø!<--- RSRC 4 H
          MSPA Ø H <---!2Ø          29!<--- RSRC 5 H
          RSPA 3 H <---!21          28!<--- LIT Ø H
          MSPA 3 H <---!22          27!<--- RSRC Ø H
          RSRC 2 H --->!23          26!---> RCS IPR L
           SPW M L --->!24          25!---> RCS TMP L
                          ---------------
```

SPA FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.24 SHIFTER CONTROL CHIP (SRK-DC614)

1. GENERAL DESCRIPTION:

The main purpose of the SRK chip is to decode the ROT field and to send the SRM chips the appropriate controls.

In addition, the SRK also outputs, on a per rotator function basis, two status signals used for microbranches.

SRK Table 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	ROT 3 H		GA1TNF
2	ROT Ø H		GA1TNF
3	ROT 5 H		GA1TNF
4	SEC FUNCTION 3 L		GA1TTN
5	D SIZE Ø H		GA1TNF
6	D SIZE 1 H		GA1TNF
7	PRI FUNCTIONS Ø L		GA1TTN
8	WMUXZ B1 H		GA1TNF
9	ROT 4 H		GA1TNF
1Ø	WMUXZ BØ H		GA1TNF
11	SHF POS 1 L		GA1TTN
14	WMUXZ B3 H		GA1TNF
15	WMUXZ B2 H		GA1TNF
16	WBUS Ø5 H		GA1TNF
17	WBUS Ø2 H		GA1TNF
18	WBUS Ø7 H		GA1TNF
19	WBUS Ø3 H		GA1TNF
2Ø	WBUS ØØ H		GA1TNF
21	WBUS Ø6 H		GA1TNF
22	WBUS Ø4 H		GA1TNF
23	SBUS Ø6 H		GA1TZF
24	SBUS Ø5 H		GA1TZF
25	WBUS Ø1 H		GA1TNF
26	SBUS Ø7 H		GA1TZF
27	SBUS ØØ H		GA1TZF
28	SBUS Ø1 H		GA1TZF
29	SBUS Ø3 H		GA1TZF
3Ø	SBUS Ø2 H		GA1TZF
31	SBUS Ø4 H		GA1TZF
32	SHF POS Ø L		GA1TTN
33	SHF POS 3 L		GA1TTN
34	SHF POS 4 L		GA1TTN
36	STATUS Ø H		GA1TTN
37	STATUS 1 H		GA1TTN
39	SHF POS 2 L		GA1TTN
4Ø	QD CLK L		GA1TNF
41	SEC FUNCTION 1 L		GA1TTN
42	SEC FUNCTION Ø L		GA1TTN
43	SEC FUNCTION 2 L		GA1TTN
44	SEC FUNCTION 5 L		GA1TTN
45	SEC FUNCTION 4 L		GA1TTN
46	ROT 1 H		GA1TNF
47	PRI FUNCTION 1 L		GA1TTN
48	ROT 2 H		GA1TNF

2 Performance:

2.1 Chip I/O Pin Summary:

NAME	I/O	PIN COUNT	ELECTRICAL	LOGICAL
WBUS <07:00> H	IN	8	TRI-STATE	ACTIVE HI
SBUS <07:00> H	IN/OUT	8	TRI-STATE	ACTIVE HI
ROT <5:0> H	IN	6	TRI-STATE	ACTIVE HI
PRI FUNCTION <1:0> L	OUT	2	TOTEM POLE	ACTIVE LOW
SEC FUNCTION <5:0> L	OUT	6	TOTEM POLE	ACTIVE LOW
SHF POS <4:0> L	OUT	5	TOTEM POLE	ACTIVE LOW
D SIZE <1:0> H	IN	2	TOTEM POLE	ACTIVE HI
WMUXZ B<3:0> H	IN	4	OPEN COLLECTOR	ACTIVE HI
STATUS <1:0> H	OUT	2	TOTEM POLE	ACTIVE HI
QD CLK L	IN	1	TOTEM POLE	(CLOCK)

NOTE: Unless otherwise specified, all signals are expressed in logical values, and all numbers are in decimals.

2.2 Output Control Pins: The SRK sends three groups of control signals to the SRM chips. These are briefly described here, with more detail in Section 2.6.

2.2.1 PRI (PRI FUNCTION <1:0> L): A two bit signal which indicates the primary function:

PRI	PRIMARY FUNCTION
0	EXTZ M,R
1	EXTZ M,M
2	EXTZ R,R
3	MISC

2.2.2 SEC (SEC FUNCTION <5:0> L): A six bit signal which indicates either the end bit position of an EXTZ primary function; or the secondary control for a MISC primary function.

2.2.3 SHF POS (SHF POS <4:0> L): A five bit signal which indicates either the starting bit position for an EXTZ function or the shift count for some MISC functions. Only bit<4:2> are used by the SRM chips. The remaining two bits are used by the ALP chips to further shift the SRM outputs by 0, 1, 2 or 3 bits.

In addition to the above three groups of signals, the SRK also outputs a two bit status signal used for micro branches. The STATUS signals are completely specified in Tables V and VI.

2.3 **Input Control Pins:** The SRK receives the following signals for controlling purposes.

2.3.1 **WMUXZ B<3,2,1,0> H:** These are four signals sent by the ALP chips to indicate whether each of the four bytes is zero:

WMUXZ B<n> H	WMUX byte n = 0
1	yes
0	no

2.3.2 **D SIZE (D SIZE <1:0> H):** These are two signals from the micro-sequencer indicating the data size as specified by the ISTRM and the D TYPE fields from the micro-word.

2.3.3 **ROT <5:0> H:** These are the six bit ROT field from the micro-word to control the rotator in general.

2.3.4 **CLOCK SIGNAL:** The only clock signal received is the QD CLK L, which is used to latch the PL and the SL.

2.4 **DATA BUS'es:** The SRK interfaces with the low order byte of the two data buses on the CPU main data path. These are the WBUS and the SBUS.

Both buses are used by the SRK for loading the two internal latches and for decoding some status conditions.

The SBUS is also used to unload the PL and the SL back onto the data path, under this condition the SRM chip will output all zero's onto the higher bytes of the SBUS.

2.5 **BLOCK DIAGRAM:** Figure 2-105 of the VAX-11/750 CPU Technical Description depicts the functional block diagram of the SRK Chip.

2.5.1 **POSITION LATCH (PL) AND SIZE LATCH (SL):** The PL and SL are 6 bit feed-through latches. In general, the SL is used to specify the size of bit field and PL the number of bits to be shifted in most of the rotator functions.

Loading of either latches is enabled explicitly by the ROT field. When enabled, data will feed through the latch as long as the clock signal "QD CLK L" is at a low level; at the rising edge of the clock, the input data is retained.

Both latches can be output to the SBUS<7:0>, with SBUS<7:6> = 0.

2.5.2 **PRI LOGIC:** The PRI logic decodes the ROT field and the D SIZE signals to output the PRI signals to the SRM chips.

2.5.3 **SEC LOGIC:** Depending on the ROT field, the SEC logic generates the six bit SEC signals from two sources:

2.5.3.1 By pure decoding of the ROT field and the D SIZE, this is used for the majority of the rotator functions.

When the PRI logic indicates a MISC function (i.e. PRI <1:0> = 3), only SEC<3:0> are meaningful. Under this condition, two numbers are listed in SRK Table 2 under the column SEC<5:0>. The first number indicates what happens to be output from SEC<5:4>, and the second number indicates SEC<3:0>, the secondary function code.

2.5.3.2 By performing the operation SL + PL<1:0> -1, this is used for some EXTZ functions which use SL to specify the size of a bit field.

2.5.4 **SHF POS LOGIC:** The SHF POS logic generates the five bit SHF POS signals from two sources:

2.5.4.1 By pure decoding of the ROT field, the D SIZE and the WMUXZ B<3:0> inputs. This is used when a shift count is explicitly specified with the rotator function.

2.5.4.2 By performing some 2's complement ALU functions on the PL and SL. The possible ALU functions are:

PL
SL
-(PL + SL)
-(PL)
PL + SL

2.5.5 **FIND FIRST LOGIC:** This is provided for the "PL <-- MSS" function, the result of which is a function of the WMUXZ B<3:0> signals and the SBUS<7:0>.

2.5.6 **STATUS:** The STATUS logic outputs a two bit signal as a function of all input control signals as well as the PL and the SL. See SRK Tables.

2.6 **Output Control Pins, Detail Description:** On a per function basis, the logical value for the PRI, SEC and SHF POS signals are listed in SRK Table 2. From the way that these three groups of signals are derived, the 64 rotator functions can be divided into eight classes, which are described herein.

2.6.1 **EXTZ, SL DEPENDENT:** For this class of functions, SL is used to specify the size of the bit field. The SEC signals are calculated as:

SEC = f(SL + PL<1:0> -1), which essentially indicates the end bit pdsition with respect to the SBUS output. The reason that only PL<1:0> is involved is that data seen at the SBUS has already been rotated by PL<4:2> nibbles.

To save decoding hardware in the SRM chips, the SRK also performs a logical operatior on output SEC<4>, so that,

let TMP= SL + PL<1:0> -1
then SEC<4> = TMP<4> .OR. TMP<5>
and SEC<5, 3:0> = TMP<5, 3:0>.

2.6.2 **EXTZ, SL INDEPENDENT:** For this class of functions, the size is explicitly specified with the rotator function, so that the SEC signals is a direct decoding of the ROT field.

2.6.3 **ROTATE RIGHT:** A function in this class is essentially the same as an EXTZ function except that the size is defaulted to 32. To achieve a rotate right functionality, the SRK forces the SEC signals to some numbers greater than 33.

2.6.4 **ROTATE LEFT:** A rotate left functionality is essentially an EXTZ function with the SIZE defaulted to 32 and with the SHF POS = the negated value of the original rotate count.

To achieve the rotate left functionality, the SRK forces the SEC' signals to some numbers greater than 33, and outputs (0 - the original rotate count) to the SHF POS signals.

2.6.5 **SHIFT RIGHT/SHIFT LEFT:** A shift left or a shift right functionality is performed in the SRM chips as a distinct function (not a subset of the EXTZ functions). The main role of the SRK is to compute the shift count.

For a shift right functionality, the shift count implied in the ROT field is outputted to the SHF POS. For a shift left functionality, the SHF POS is the negated value of the shift count. An exception to this class is the ASL.R.SIZ function, which involves more than a shift left functionality.

2.6.5.1 **ASL.R.SIZ:** When SHF POS is 0, an ASL R function in the SRM chip would produce all zero's. To remedy this deficiency, the SRK would send the SRM chips a different set of signals when D SIZE = 0.

Specifically, when D SIZE = 0:

PRI = 2 (EXTZ R,R)
SEC = 52 (have the same effect as SIZE = 32)
SHF POS = 0.

And when D SIZE = 1, 2, 3:

D SIZE	PRI	SEC	SHF POS
1	3(MISC)	52(ASL R)	31
2	3(MISC)	52(ASL R)	30
3	3(MISC)	52(ASL R)	29

2.6.6 LATCH LOADING: When either the PL or the SL is loaded, the SRK will control the SRM chips so that meaningful data appear on the SBUS. In particular, when data is loaded from the WBUS, (i.e. SL.PL __ WB or PL.SL __ WB), the SRM chips will perform a "Low byte OFF" function so that the SRK can output the other latch (the one which is not being loaded) onto the SBUS. When data is loaded from the SBUS (i.e. OLIT0.PL __ LIT or OLIT0.SL __ LIT), the SRM chips will perform a 'OLIT0" function so that the micro-word short literal is loaded into the latch.

In addition, two other functions are provided to load the PL for special purpose. These are (1) OLIT0.PL43 WB and (2) PL <-- MSS functions.

2.6.6.1 OLIT0.PL 43 <-- WB: PL<4:3> is loaded with WB<1:0>, other bits of PL are unchanged. The control signals received by the SRM chips is a OLIT0 function so that data appearing on the SBUS is the 1-extended version of the LITRL field.

2.6.6.2 PL <-- MSS: For this function, the SRK examines the WMUXZ signals to locate the left most non zero byte on the MBUS, and then controls the SRM chips to rotate that byte onto SBUS<7:0>. With the information on the SBUS<7:0> and the WMUXZ signals, the SRK is able to locate the most significant bit set on the MBUS. Signals sent to the SRM chips are:

PRI = 1 (EXTZ M,M)
SEC = 63 (SIZE = 32)
SHF POS = M'N'000(BIN) where M'N is a function of the WMUXZ signals.

WMUXZ<3:0>	M	N
0XXX	1	1
10XX	1	0
110X	0	1
111X	0	0

Also PL<4:3> is loaded with M'M and PL<2:0> loaded with a value decoded from SBUS<7:0>:

SBUS<7:0>	PL<2:0>
1XXXXXXX	111
01XXXXXX	110
001XXXXX	101
0001XXXX	100
00001XXX	011
000001XX	010
0000001X	001
0000000X	000

2.6.7 **LATCH UNLOADING:** Both the PL and SL can be read onto the SBUS. The SRK controls the SRM chips such that they turn off the low byte and output all zeros to the higher bytes.

2.6.8 **MISCELLANEOUS:** Functions that do not fall into the above classes are more or less some types of data shuffling operations handled directly by the SRM chips. The role of the SRK is merely to decode the ROT field and to send the SRM chips the appropriate signals.

SRK Table 2

ROT	MNEMONIC	PRI<1:0>	SEC<5:0>	SHF POS<4:0>	CLASS
0	XZ.MR	0(EXTZ M,R)	See 2.6.1	PL	1
1	XZ.MM	1(EXTZ M,M)	See 2.6.1	PL	1
2	XZ.RR	2(EXTZ R,R)	See 2.6.1	PL	1
3	ASR.M.P	3(MISC)	3,12(ASR M)	PL	5
4	RR.MR.P	0(EXTZ M,R)	63	PL	3
5	RR.MM.P	1(EXTZ M,M)	63	PL	3
6	RR.RR.P	2(EXTZ R,R)	63	PL	3
7	RR.MR.S	0(EXTZ M,R)	63	SL	3
8	RL.RM.4	0(EXTZ M,R)	60	28	4
9	RR.MR.4	0(EXTZ M,R)	60	4	3
10	RR.RR.SIZ	2(EXTZ R,R)	60	(DSIZE+1)*8	3

SRX Table 2 (Cont)

ROT	MNEMONIC	PRI<1:0>	SEC<5:0>	SHF POS<4:0>	CLASS
11	RR.MR.9	0(EXTZ M,R)	62	9	3
12	XZ.PTX	1(EXTZ M,M)	25	7	2
13	XZ.VPN	1(EXTZ M,M)	21	9	2
14	RR.MM.SIZ	1(EXTZ M,M)	57	(DSIZE+1)*8	3
15	GETNIB	1(EXTZ M,M)	3	0	2
16	GETEXP	1(EXTZ M,M)	10	7	2
17	RL.MM.PTE	1(EXTZ M,M)	58	23	4
18	CLR2BM	3(MISC)	3,1(CLR 2 BYTE)	0	8
19	CLR1BM	3(MISC)	3,0(CLR 1 BYTE)	16	8
20	CLR3BM	3(MISC)	0,3(CLR 3 BYTE)	0	8
21	ASL.R.7	3(MISC)	2,4(ASL R)	25	5
22	ZERO	3(MISC)	3,5(ASL M)	0	5
23	ASL.R.SIZ	(S E E	2.6.5.1	)	5
24	BCDSWP	3(MISC)	0,9(BCD SWAP)	0	8
25	GETFPF	3(MISC)	3,8(FP FRACT)	1	8
26	FPACK	3(MISC)	2,11(FP PACK)	1	8
27	CVTPN	3(MISC)	2,10(CVTPN)	0	8
28	CONX.SIZ	3(MISC)	0,13(CONSTANT 8)	(3 - DSIZE)	8
29	ASR.M.3	3(MISC)	0,12(ASR M)	3	5
30	FPLIT	3(MISC)	2,15(FP LIT)	0	8
31	CVTNP	3(MISC)	0,14(CVTNP)	0	8
32	RL.RM.PS	0(EXTZ M,R)	63	-(PL + SL)	4
33	RL.MM.P	1(EXTZ M,M)	63	-(PL)	4
34	RL.RR.P	2(EXTZ R,R)	63	-(PL)	4

SRK Table 2 (Cont)

ROT	MNEMONIC	PRI<1:0>	SEC<5:0>	SHF POS<4:0>	CLASS
35	RL.RM.P	0(EXTZ M,R)	63	-(PL)	4
36	RR.MR.PS	0(EXTZ M,R)	63	(PL + SL)	3
37	RR.MM.PS	1(EXTZ M,M)	63	(PL + SL)	3
38	RR.RR.PS	2(EXTZ R,R)	63	(PL + SL)	3
39	PL <-- MSS	1(EXTZ M,M)	63	SEE 2.6.6.2	6
40	ASL.R.P	3(MISC)	3,4(ASL R)	-(PL)	5
41	ASL.M.P	3(MISC)	3,5(ASL M)	-(PL)	5
42	ASR.M.-P	3(MISC)	3,12(ASR M)	-(PL)	5
43	ZLITPL	3(MISC)	3,7(LIT ZERO)	-(PL)	8
44	PL	3(MISC)	1,2(LO BYTE OFF)	20	7
45	PL.SL __ WB	3(MISC)	1,2(LO BYTE OFF)	8	6
46	SL	3(MISC)	3,2(LO BYTE OFF)	24	7
47	SL.PL __ WB	3(MISC)	0,2(LO BYTE OFF)	SAME AS ROT=39	6
48	ZLIT0	3(MISC)	0,7(LIT ZERO)	0	8
49	ZLIT28	3(MISC)	3,7(LIT ZERO)	4	8
50	ZLIT24	3(MISC)	3,7(LIT ZERO)	8	8
51	ZLIT20	3(MISC)	3,7(LIT ZERO)	12	8
52	ZLIT16	3(MISC)	0,7(LIT ZERO)	16	8
53	ZLIT12	3(MISC)	2,7(LIT ZERO)	20	8

SRK Table 2 (Cont)

ROT	MNEMONIC	PRI<1:0>	SEC<5:0>	SHF POS<4:0>	CLASS
54	ZLIT8	3(MISC)	3,7(LIT ZERO)	24	8
55	ZLIT4	3(MISC)	3,7(LIT ZERO)	28	8
56	OLIT0	3(MISC)	0,6(LIT ONE)	0	8
57	MINUS1	3(MISC)	3,6(LIT ONE)	0	8
58	OLIT24	3(MISC)	2,6(LIT ONE)	8	8
59	OLIT0.PL__LIT	3(MISC)	2,6(LIT ONE)	0	6
60	OLIT16	3(MISC)	0,6(LIT ONE)	16	8
61	OLIT0.SL__LIT	3(MISC)	0,6(LIT ONE)	0	6
62	OLIT8	3(MISC)	2,6(LIT ONE)	24	8
63	OLIT0.PL43__WB	3(MISC)	0,6(LIT ONE)	0	6

SRK Table 3

ROTATOR FUNCTIONS

ROT	MNEMONIC	DESCRIPTION	STATUS<1>	STATUS<0>
0	XZ.MR	;EXTZ M'R, POS=PL, SIZE=SL ;	SL<5:0>=0	(PL<4:0>+SL)>32
1	XZ.MM	;EXTZ M'M, POS=PL, SIZE=SL ;	SL<5:0>=0	(PL<4:0>+SL)>32
2	XZ.RR	;EXTZ R'R, POS=PL, SIZE=SL ;	SL<5:0>=0	(PL<4:0>+SL)>32
3	ASR.M.P	;ARI SHF RIGHT M,NO.BITS=PL;	0	PL<5>
4	RR.MR.P	;ROT RIGHT M'R, NO.BITS=PL ;	SL<5:0>=0	(PL<4:0>+SL)>32
5	RR.MM.P	;ROT RIGHT M'M, NO.BITS=PL ;	SL<5:0>=0	(PL<4:0>+SL)>32
6	RR.RR.P	;ROT RIGHT R'R, NO.BITS=PL ;	SL<5:0>=0	(PL<4:0>+SL)>32
7	RR.MR.S	;ROT RIGHT M'R, NO.BITS=SL ;	0	PL<5>
8	RL.RM.4	;ROT LEFT R'M, NO. BITS=4 ;	D size<1>	D size<0>
9	RR.MR.4	;ROT RIGHT M'R, NO.BITS=4 ;	D size<1>	D size<0>
10	RR.RR.SIZ	;ROT RIGHT R'R, BY 1,2,3,0 BYTES ;	D size<1>	D size<0>
11	RR.MR.9	;ROT RIGHT M'R, NO.BITS=9 ;	D size<1>	D size<0>
12	XZ.PTX	;EXTZ M'M, POS=07,SIZE=23 ;	D size<1>	D size<0>

SRK Table 3 (Cont)

ROTATOR FUNCTIONS

ROT	MNEMONIC	DESCRIPTION		STATUS<1>	STATUS<0>
13	XZ.VPN	;EXTZ M'M, POS-09,SIZE=21	;	D size<1>	D size<0>
14	RR.MM.SIZ	;ROT RIGHT M'M BY 1,2,3,0 BYTES	;	D size<1>	D size<0>
15	GETNIB	;GET 0'MBUS<3:0>	;	D size<1>	D size<0>
16	GETEXP	;EXTZ M'M POS=7, SIZE = 8	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
17	RL.MM.PTE	;ROT LEFT M'M, NO. BITS=9	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
18	CLR2BM	;CLR M<15:0>	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
19	CLR1BM	;CLR M<07:0>	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
20	CLR3BM	;CLR M<23:0>	;	ASCII sign	WB.ne.45
21	ASL.R.7	;ARI SHF LEFT R BY 7 BITS	;	ASCII sign	WB.ne.45
22	ZERO	;CONSTANT 0	;	ASCII sign	WB.ne.45
23	ASL.R.SIZ	;ARI SHF LEFT R BY 0,1,2,3 BITS	;	ASCII sign	WB.ne.45
24	BCDSWP	;BCD SWAP, MBUS	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
25	GETFPF	;UNPACK FP FRACTION, M'R	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
26	FPACK	;PACK FP DATUM,M=FRAC R=EXP	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
27	CVTPN	;CONVERT PACKED TO NUMERIC	;	SB<3:0>.ne.0	SB<3:0>.ne.(11,13)
28	CONX.SIZ	;CONSTANT 1,2,4,8 ON SIZE	;	ASCII sign	WB.ne.45
29	ASR.M.3	;ARI SHF RIGHT M,NO.BITS=3	;	ASCII sign	WB.ne.45
30	FPLIT	;EXPAND FLOATING POINT LIT	;	ASCII sign	WB.ne.45
31	CVTNP	;CONVERT NUMERIC TO PACKED	;	ASCII sign	WB.ne.45

NOTE: WB denotes WBUS low byte
SB denotes SBUS
"ASCII sign" denotes WB.ne. (32, 43, or 45)

SRK Table 4

ROTATOR FUNCTIONS

ROT	MNEMONIC	DESCRIPTION	STATUS<1>	STATUS<0>
32	RL.RM.PS	;ROT LEFT R'M,NO.BITS=PL+SL;	SL<5:0>=0	undefined
33	RL.MM.P	;ROT LEFT M'M,NO. BITS = PL;	SL<5:0>=0	PL<5>
34	RL.RR.P	;ROT LEFT R'R,NO. BITS = PL;	SL<5:0>=0	PL<5>
35	RL.RM.P	;ROT LEFT P'M,NO. BITS = PL;	WMUX<31:16> .ne.0	WMUX<15:00> .ne.0
36	RR.MR.PS	;ROT RIGHT M'R,NO.BITS=PL+SL;	SL<5:0>=0	(PL<4:0>+SL)>32
37	RR.MM.PS	;ROT RIGHT M'M,NO.BITS=PL+SL;	SL<5:0>=0	(PL<4:0>+SL)>32
38	RR.RR.PS	;ROT RIGHT R'R,NO.BITS=PL+SL;	SL<5:0>=0	(PL<4:0>+SL)>32
39	PL_MSS	;FIND MOST SIG. BIT SET MBUS;	WMUX<31:16> .ne.0	WMUX<15:00> .ne.0
40	ASL.R.P	;ARI SHF LEFT R, NO.BITS=PL ;	PL<4:0>=0	pl<5>
41	ASL.M.P	;ARI SHF LEFT M. NO.BITS=PL ;	PL<4:0>=0	pl<5>
42	ASR.M.-P	;ARI SHF RIGHT M,NO.BTIS=-PL;	PL<4:0>=0	pl<5>
43	ZLITPL	;0 EXT LIT&ROT left PL BITS ;	PL<4:0>=0	0
44	PL	;SBUS <- PL ;	WB > 31	WB.ne.[1,32]
45	PL.SL_WB	;SL <- WBUS<5:0>, SBUS <- PL;	WB > 31	WB.ne.[1,32]
46	SL	;SBUS <- SL ;	WB > 31	WB.ne.[1,32]
47	SL.PL_WB	;PL <- WBUS<5:0>, SBUS <- SL;	WB > 31	WB.ne.[1,32]
48	ZLIT0	;0 EXT LIT&ROT left 00 BITS;	WB<7>=0	|WB| .ge. 32
49	ZLIT28	;0 EXT LIT&ROT left 28 BITS;	WB<7>=0	|WB| .ge. 32
50	ZLIT24	;0 EXT LIT&ROT left 24 BITS;	WB<7>=0	|WB| .ge. 32
51	ZLIT20	;0 EXT LIT&ROT left 20 BITS;	WB<7>=0	|WB| .ge. 32
52	ZLIT16	;0 EXT LIT&ROT left 16 BITS;	WB<7>=0	|WB| .ge. 32
53	ZLIT12	;0 EXT LIT&ROT left 12 BITS;	WB<7>=0	|WB| .ge. 32
54	ZLIT8	;0 EXT LIT&ROT left 08 BITS;	WB<7>=0	|WB| .ge. 32
55	ZLIT4	;0 EXT LIT&ROT left 04 BITS;	WB<7>=0	|WB| .ge. 32
56	OLIT0	;1 EXT LIT&ROT left 00 BITS;	WB<7>=0	|WB| .ge. 32
57	MINUS1	;CONSTANT OF ALL 1'S ;	WB<7>=0	|WB| .ge. 32
58	OLIT24	;1 EXT LIT&ROT left 24 BITS;	WB<7>=0	|WB| .ge. 32
59	OLIT0.PL_LIT	;PL <- LIT ;	WB<7>=0	|WB| .ge. 32
60	OLIT16	;1 EXT LIT&ROT left 16 BITS;	WB<7>=0	|WB| .ge. 32
61	OLIT0.SL_LIT	;SL <- LIT ;	WB<7>=0	|WB| .ge. 32
62	OLIT8	;1 EXT LIT&ROT left 08 BITS;	WB<7>=0	|WB| .ge. 32
63	OLIT0.PL43_WB	;PL<4:3> <- WBUS<1:0> ;	WB<7>=0	|WB| .ge. 32

Note: |WB| Denotes the magnitude of a 2's complement number on WBUS<7:0>.
[1,32] Denotes the Range of 1 thru 32.

```
                         ---------------
          ROT 3 H --->!01          48!<--- ROT 2 H
          ROT 0 H --->!02          47!---> PRI FUNCTION 1 L
          ROT 5 H --->!03          46!<--- ROT 1 H
 SEC FUNCTION 3 L <---!04          45!---> SEC FUNCTION 4 L
       D SIZE 0 H --->!05          44!---> SEC FUNCTION 5 L
       D SIZE 1 H --->!06          43!---> SEC FUNCTION 2 L
 PRI FUNCTION 0 L <---!07          42!---> SEC FUNCTION 0 L
      WMUXZ B1 H --->!08          41!---> SEC FUNCTION 1 L
          ROT 4 H --->!09          40!<--- QD CLK L
      WMUXZ B0 H --->!10 ....... 39!---> SHF POS 2 L
     SHF POS 1 L <---!11 .     . 38!---- GROUND
              VGA ----!12 . LID . 37!---> STATUS 1 H
              VCC ----!13 . DOWN. 36!---> STATUS 0 H
      WMUXZ B3 H --->!14 .     . 35!---- GROUND
      WMUXZ B2 H --->!15 ....... 34!---> SHF POS 4 L
       WBUS 05 H --->!16          33!---> SHF POS 3 L
       WBUS 02 H --->!17          32!---> SHF POS 0 L
       WBUS 07 H --->!18          31!<--> SBUS 04 H
       WBUS 03 H --->!19          30!<--> SBUS 02 H
       WBUS 00 H --->!20          29!<--> SBUS 03 H
       WBUS 06 H --->!21          28!<--> SBUS 01 H
       WBUS 04 H --->!22          27!<--> SBUS 00 H
       SBUS 06 H <-->!23          26!<--> SBUS 07 H
       SBUS 05 H <-->!24          25!<--- WBUS 01 H
                         ---------------
```

SRK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.25 SHIFT AND ROTATING MULTIPLEXER (SRM-DC613)

1. GENERAL DESCRIPTION:

The SRM chip contains logic to perform multiple bit shifting variable length bit field extraction, constant generation, and some bit-shuffling type operations useful in executing the VAX instruction set.

SRM TABLE 1
Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	SHF POS 2 L		GA1TNG
2	SHF POS 4 L		GA1TNG
3	SHF POS 3 L		GA1TNG
4	LIT 8 H		GA1TNG
5	LIT 4 H		GA1TNG
6	SEC FUNCTION 4 L		GA1TNG
7	SEC FUNCTION 3 L		GA1TNG
8	PRI FUNCTION 1 L		GA1TNG
9	LIT 0 H		GA1TNG
10	ID 0 L		GA1TNF
11	ID 1 L		GA1TNF
14	PRI FUNCTION 0 L		GA1TNG
15	SB 12 H		GA1TTN
16	SB 08 H		GA1TTN
17	SEC FUNCTION 1 L		GA1TNG
18	SEC FUNCTION 2 L		GA1TNG
19	SEC FUNCTION 0 L		GA1TNG
20	SB 00 H		GA1TZN
21	SB 04 H		GA1TZN
22	SB 16 H		GA1TTN
23	SB 20 H		GA1TTN
24	SB 24 H		GA1TTN
25	SB 28 H		GA1TTN
26	SB 32 H		GA1TTN
27	MB 00 L		GA1TNH
28	MB 20 L		GA1TNH
29	MB 04 L		GA1TNH
30	MB 12 L		GA1TNH
31	MB 08 L		GA1TNH
32	MB 16 L		GA1TNH
33	MB 24 L		GA1TNH
34	MB 28 L		GA1TNH
36	PHASE H		GA1TNG
37	RB 04 L		GA1TPF
39	RB 00 L		GA1TPF
40	RB 08 L		GA1TPF
41	RB 12 L		GA1TPF
42	RB 24 L		GA1TPF
43	RB 16 L		GA1TPF
44	RB 20 L		GA1TPF
45	RB 28 L		GA1TPF
46	MBUS 31 L		GA1TNG
47	MB XX L		GA1TNG
48	SEC FUNCTION 5 L		GA1TNG

2. Performance Requirements:

2.1 Interface: Input data operated on by the SRM chips come from three sources:

A. MBUS
B. RBUS
C. LITERAL field from the microword,

Data from the MBUS and the RBUS are internally latched during the first half of the cycle. The clock used is the "PHASE H" signal. When Phase H is at a low level, input data is fed through the latch. At the rising edge of PHASE H, input data is retained until PHASE H goes low again.

Results from the SRM chips are output to the SBUS of the CPU main Data Path.

For pin out reasons, data input/output to/from the SRM chips are 4-way interleaved such that each chip contains one bit of a nibble.

2.2 Chip I/O Summary: Per Table 2. See Para. 2.3

SRM TABLE 2

CHIP I/O SUMMARY DATA

NAME	I/O	ELECTRICAL	NOTE
MB <n> L	Input	N/A	n = 28, 24, 20, 16, 12, 08, 04, 00
RB <n> L	(See 2.3.2)		n = 28, 24, 20, 16, 12, 08, 04, 00
LIT <n> H	Input	N/A	n = 8, 4, 0
MBUS 31 L	Input	N/A	
PHASE H	Input	N/A	
PRI Function <n> L	Input	N/A	n = 1, 0
SEC Function <n> L	Input	N/A	n = 5, 4, 3, 2, 1, 0
SHF POS <n> L	Input	N/A	n = 4, 3, 2
SB <n> H	Output	(See 3.3.3.10)	n = 32, 28, 24, 20, 16, 12, 08, 04, 00
ID <n> L	Input	N/A	n = 1, 0
MB XX L	Input	N/A	

2.3 **Pin Functions:**

2.3.1 **MB <n> L:** The 8 input data from the MBUS. The MB input pins are named as MB <28,24,2Ø,16,12,Ø8,Ø4,ØØ> L.

2.3.2 **RB <n> L:** Same as MBUS except that data is from the RBUS. The "RB <n> L" pins are used as inputs. Pull up resistors are provided by the chip so that unless the data is at a low level, signals on these pins are pulled up to a high level.

2.3.3 **LIT <8, 4, Ø> H:** Data input from the literal field of the microword.

2.3.4 **MBUS 31 L:** An input serves as the shift-in for the "arithmetic shift right MBUS" function (ASR M).

2.3.5 **PHASE H:** When PHASE H = L, data from the MBUS and the RBUS are internally latched inside the SRM chips. These include the 8 input data from the MBUS and RBUS respectively, the "MB XX" and the single "MBUS<31>" data.

2.3.6 **PRI (PRI FUNCTION <1:Ø> L):** The two control inputs which determine the primary functions and hence also the interpretation of another group of pins, the "SEC" pins.

PRI	PRIMARY FUNCTION NAME	
H H	EXTZ	M,R
H L	EXTZ	M,M
L H	EXTZ	R,R
L L	MISC.	See 2.3.7

All these functions are specified in SRM Table 3.

2.3.7 **SEC:** These are the six control inputs, the interpretation of which depends on the primary function. When the primary function is of the EXTZ type, the SEC pins and the ID pins (2.3.9) specify if a SBUS output pin is forced to zero, regardless the value of the other data inputs. See SRM Tables 3 and 4. When the primary function specifies MISC, SEC<3:Ø> serve as the secondary function control pins. The 16 secondary functions are:

SEC<3:Ø>	SECONDARY FUNCTION NAME
HHHH(Ø)	CLR 1 BYTE
HHHL(1)	CLR 2 BYTE
HHLH(2)	LO BYTE OFF
HHLL(3)	CLR 3 BYTE
HLHH(4)	ASL R
HLHL(5)	ASL M
HLLH(6)	LIT ONE
HLLL(7)	LIT ZERO
LHHH(8)	FP FRACT
LHHL(9)	BCD SWAP
LHLH(1Ø)	CVTPN
LHLL(11)	FP PACK
LLHH(12)	ASR M
LLHL(13)	CONSTANT 8
LLLH(14)	CVTNP
LLLL(15)	FP LIT

The functions are defined in SRM Table 3.

2.3.8 POS: The interpretation of these three input pins is also function dependent. Refer to SRM Table 3. For the EXTZ functions, POS specifies the starting nibble position of a bit field to be extracted:

POS	STARTING NIBBLE POSITION
HHH	Ø
HHL	1
HLH	2
HLL	3
LHH	4
LHL	5
LLH	6
LLL	7

For the ASR M function, POS specifies the number of positions to be shifted right:

POS	NIBBLES TO BE SHIFTED
HHH	Ø
HHL	1
HLH	2
HLL	3
LHH	4
LHL	5
LLH	6
LLL	7

For the ASL M and ASL R functions, POS specifies the number of positions to be shifted left:

POS	NIBBLES TO BE SHIFTED
HHH	8
HHL	7
HLH	6
HLL	5
LHH	4
LHL	3
LLH	2
LLL	1

For all other ʟunctions, POS is not used.

2.3.9 ID <1:0> L: To perform some functions it is necessary to identify the slice position which a chip is hooked up to. This is done via two ID pins, which will be hardwired to give each of the SRM slices a distinct identity. See Tables SRM 3 and 4.

2.3.10 SB <n> H: The nine output pins to the SBUS.

Output pin SB<04,00> are tri-state and the remaining pins totem pole.

2.3.11 FUNCTION TABLE: The 19 functions (3 EXTZ functions plus 16 secondary functions) performed by the SRM chip are specified in Table IV, which express the output from each of the SB pins in terms of:

1. function name
2. POS (SHF POS <4:2> L), whenever applicable
3. input data.

Notations used in the Table are:

Mi = latched data from the input pin MB i L,
Ri = latched data from the input pin RB i L,
Li = data from the input pin LIT i H,
SBi = output pin SB i H,
Z = high impedance state
MXX = latched data from the input pin MB XX L

In addition, all conditions which involve the ID pins are also marked.

SRM TABLE 3

FUNCTION TABLE

Function	POS	SB32	SB28	SB24	SB20	SB16	SB12	SB08	SB04	SB00
EXTZ M,R(1)	HHH	M00	R28	R24	R20	R16	R12	R08	R04	R00
	HHL	M04	M00	R28	R24	R20	R16	R12	R08	R04
	HLH	M08	M04	M00	R28	R24	R20	R16	R12	R08
	HLL	M12	M08	M04	M00	R28	R24	R20	R16	R12
	LHH	M16	M12	M08	M04	M00	R28	R24	R20	R16
	LHL	M20	M16	M12	M08	M04	M00	R28	R24	R20
	LLH	M24	M20	M16	M12	M08	M04	M00	R28	R24
	LLL	M28	M24	M20	M16	M12	M08	M04	M00	R28
EXTZ M,M(1)	HHH	M00	M28	M24	M20	M16	M12	M08	M04	M00
	HHL	M04	M00	M28	M24	M20	M16	M12	M08	M04
	HLH	M08	M04	M00	M28	M24	M20	M16	M12	M08
	HLL	M12	M08	M04	M00	M28	M24	M20	M16	M12
	LHH	M16	M12	M08	M04	M00	M28	M24	M20	M16
	LHL	M20	M16	M12	M08	M04	M00	M28	M24	M20
	LLH	M24	M20	M16	M12	M08	M04	M00	M28	M24
	LLL	M28	M24	M20	M16	M12	M08	M04	M00	M28
EXTZ R,R(1)	HHH	R00	R28	R24	R20	R16	R12	R08	R04	R00
	HHL	R04	R00	R28	R24	R20	R16	R12	R08	R04
	HLH	R08	R04	R00	R28	R24	R20	R16	R12	R08
	HLL	R12	R08	R04	R00	R28	R24	R20	R16	R12
	LHH	R16	R12	R08	R04	R00	R28	R24	R20	R16
	LHL	R20	R16	R12	R08	R04	R00	R28	R24	R20
	LLH	R24	R20	R16	R12	R08	R04	R00	R28	R24
	LLL	R28	R24	R20	R16	R12	R08	R04	R00	R28
ASR M	HHH	M31	M28	M24	M20	M16	M12	M08	M04	M00
	HHL	M31	M31	M28	M24	M20	M16	M12	M08	M04
	HLH	M31	M31	M31	M28	M24	M20	M16	M12	M08
	HLL	M31	M31	M31	M31	M28	M24	M20	M16	M12
	LHH	M31	M31	M31	M31	M31	M28	M24	M20	M16
	LHL	M31	M31	M31	M31	M31	M31	M28	M24	M20
	LLH	M31	M31	M31	M31	M31	M31	M31	M28	M24
	LLL	M31	M31	M31	M31	M31	M31	M31	M31	M28

SRM TABLE 3 (CONT)

FUNCTION TABLE

Function	POS	SB32	SB28	SB24	SB20	SB16	SB12	SB08	SB04	SB00
ASL M										
	HHH	M00	0	0	0	0	0	0	0	0
	HHL	M04	M00	0	0	0	0	0	0	0
	HLH	M08	M04	M00	0	0	0	0	0	0
	HLL	M12	M08	M04	M00	0	0	0	0	0
	LHH	M16	M12	M08	M04	M00	0	0	0	0
	LHL	M20	M16	M12	M08	M04	M00	0	0	0
	LLH	M24	M20	M16	M12	M08	M04	M00	0	0
	LLL	M28	M24	M20	M16	M12	M08	M04	M00	0
ASL R										
	HHH	R00	0	0	0	0	0	0	0	0
	HHL	R04	R00	0	0	0	0	0	0	0
	HLH	R08	R04	R00	0	0	0	0	0	0
	HLL	R12	R08	R04	R00	0	0	0	0	0
	LHH	R16	R12	R08	R04	R00	0	0	0	0
	LHL	R20	R16	R12	R08	R04	R00	0	0	0
	LLH	R24	R20	R16	R12	R08	R04	R00	0	0
	LLL	R28	R24	R20	R16	R12	R08	R04	R00	0
LIT ZERO										
	HHH	L00	0	0	0	0	0	L08	L04	L00
	HHL	L04	L00	0	0	0	0	0	L08	L04
	HLH	L08	L04	L00	0	0	0	0	0	L08
	HLL	0	L08	L04	L00	0	0	0	0	0
	LHH	0	0	L08	L04	L00	0	0	0	0
	LHL	0	0	0	L08	L04	L00	0	0	0
	LLH	0	0	0	0	L08	L04	L00	0	0
	LLL	0	0	0	0	0	L08	L04	L00	0
LIT ONE$_{(4)}$										
	HHH	L00	1	1	1	1	1	L08	L04	L00
	HHL	L04	L00	1	1	1	1	1	L08	L04
	HLH	L08	L04	L00	1	1	1	1	1	L08
	HLL	1	L08	L04	L00	1	1	1	1	1
	LHH	1	1	L08	L04	L00	1	1	1	1
	LHL	1	1	1	L08	L04	L00	1	1	1
	LLH	1	1	1	1	L08	L04	L00	1	1
	LLL	1	1	1	1	1	L08	L04	L00	1

SRM TABLE 3 (CONT)

FUNCTION TABLE

Function	POS	SB32	SB28	SB24	SB20	SB16	SB12	SB08	SB04	SB00
FP FRACT	XXX	0	M04 (2)	M00	M28	M24	M20	M16	R28	R24
BCD SWAP	XXX	0	M04	M00	M12	M08	M20	M16	M28	M24
FP PACK	XXX	M24	M20	M16	M12	MXX	R04	R00	M28	M24
CVTPN	XXX	0	R16	M08	R00	M12	R04	M00	R00	M04
CVTNP	XXX	0	R16	R24	R00	R08	M16	M24	M00	M08
FP LIT	XXX	0	0	0	0	0	0 (3)	M04	M00	0
CONSTANT 8	XXX	0	0	0	0	0	0	0	0	0 (2)
CLR 1 BYTE	XXX	0	M28	M24	M20	M16	M12	M08	0	0
CLR 2 BYTE	XXX	0	M28	M24	M20	M16	0	0	0	0
CLR 3 BYTE	XXX	0	M28	M24	0	0	0	0	0	0
LO BYTE OFF	XXX	0	0	0	0	0	0	0	Z	Z

NOTES:
1. For all EXTZ functions, the SB output pins may also be forced to zero as specified in Table V.
2. Output is forced to 1 if ID=3.
3. Output is forced to 1 if ID=2.
4. When SEC FUNCTION 4 L=L, all outputs forced to H.

SRM TABLE 4

FUNCTION TABLE

	SB32	SB28	SB24	SB20	SB16	SB12	SB08	SB04	SB00	<--SB PIN #
SBC<5:0>	3210	3210	3210	3210	3210	3210	3210	3210	3210	<--ID<1:0>
HHHHHH	X111	1111	1111	1111	1111	1111	1111	1111	1110	
HHHHHL	X111	1111	1111	1111	1111	1111	1111	1111	1100	
HHHHLH	X111	1111	1111	1111	1111	1111	1111	1111	1000	
HHHHLL	X111	1111	1111	1111	1111	1111	1111	1111	0000	
HHHLHH	X111	1111	1111	1111	1111	1111	1111	1110	0000	
HHHLHL	X111	1111	1111	1111	1111	1111	1111	1100	0000	
HHHLLH	X111	1111	1111	1111	1111	1111	1111	1000	0000	
HHHLLL	X111	1111	1111	1111	1111	1111	1111	0000	0000	
HHLHHH	X111	1111	1111	1111	1111	1111	1110	0000	0000	
HHLHHL	X111	1111	1111	1111	1111	1111	1100	0000	0000	
HHLHLH	X111	1111	1111	1111	1111	1111	1000	0000	0000	
HHLHLL	X111	1111	1111	1111	1111	1111	0000	0000	0000	
HHLLHH	X111	1111	1111	1111	1111	1110	0000	0000	0000	
HHLLHL	X111	1111	1111	1111	1111	1100	0000	0000	0000	
HHLLLH	X111	1111	1111	1111	1111	1000	0000	0000	0000	
HHLLLL	X111	1111	1111	1111	1111	0000	0000	0000	0000	
HLHHHH	X111	1111	1111	1111	1110	0000	0000	0000	0000	
HLHHHL	X111	1111	1111	1111	1100	0000	0000	0000	0000	
HLHHLH	X111	1111	1111	1111	1000	0000	0000	0000	0000	
HLHHLL	X111	1111	1111	1111	0000	0000	0000	0000	0000	
HLHLHH	X111	1111	1111	1110	0000	0000	0000	0000	0000	
HLHLHL	X111	1111	1111	1100	0000	0000	0000	0000	0000	
HLHLLH	X111	1111	1111	1000	0000	0000	0000	0000	0000	
HLHLLL	X111	1111	1111	0000	0000	0000	0000	0000	0000	
HLLHHH	X111	1111	1110	0000	0000	0000	0000	0000	0000	
HLLHHL	X111	1111	1100	0000	0000	0000	0000	0000	0000	
HLLHLH	X111	1111	1000	0000	0000	0000	0000	0000	0000	
HLLHLL	X111	1111	0000	0000	0000	0000	0000	0000	0000	
HLLLHH	X111	1110	0000	0000	0000	0000	0000	0000	0000	
HLLLHL	X111	1100	0000	0000	0000	0000	0000	0000	0000	
HLLLLH	X111	1000	0000	0000	0000	0000	0000	0000	0000	
HLLLLL	X111	0000	0000	0000	0000	0000	0000	0000	0000	

SRM TABLE 4 (CONT)

FUNCTION TABLE

	SB32	SB28	SB24	SB20	SB16	SB12	SB08	SB04	SB00	<--SB PIN #
SEC<5:0>	3210	3210	3210	3210	3210	3210	3210	3210	3210	<--ID<1:0>
LLHHHH	X110	0000	0000	0000	0000	0000	0000	0000	0000	
LLHHHL	X100	0000	0000	0000	0000	0000	0000	0000	0000	
LLHHLH	X000	0000	0000	0000	0000	0000	0000	0000	0000	
LLHHLL	X000	0000	0000	0000	0000	0000	0000	0000	0000	
LLLHHL	XX00	0000	0000	0000	0000	0000	0000	0000	0000	
LLLHLH	X000	0000	0000	0000	0000	0000	0000	0000	0000	
LLLLHH	XXX0	0000	0000	0000	0000	0000	0000	0000	0000	
LLLLLH	X000	0000	0000	0000	0000	0000	0000	0000	0000	
LLLLLL	X000	0000	0000	0000	0000	0000	0000	0000	0000	
Others	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	

TABLE V

Notation:
X = Don't care case
1 = Force SBnn pin to Low for the corresponding SEC and ID combinations.
0 = Do not force SBnn type pin to Low

Mask off definions for the EXT2 functions.

```
                              ---------------
     SHF POS 2 L --->!01         48!<--- SEC FUNCTION 5 L
     SHF POS 4 L --->!02         47!<--- MB XX L
     SHF POS 3 L --->!03         46!<--- MBUS 31 L
         LIT 8 H --->!04         45!<--- RB 28 L
         LIT 4 H --->!05         44!<--- RB 20 L
SEC FUNCTION 4 L --->!06         43!<--- RB 16 L
SEC FUNCTION 3 L --->!07         42!<--- RB 24 L
PRI FUNCTION 1 L --->!08         41!<--- RB 12 L
         LIT 0 H --->!09         40!<--- RB 08 L
          ID 0 L --->!10 ....... 39!<--- RB 00 L
          ID 1 L --->!11 .     . 38!---- GROUND
             VGA ----!12 . LID . 37!<--- RB 04 L
             VCC ----!13 . DOWN. 36!<--- PHASE H
PRI FUNCTION 0 L --->!14 .     . 35!---- GROUND
        SB 12 H <----!15 ....... 34!<--- MB 28 L
        SB 08 H <----!16         33!<--- MB 24 L
SEC FUNCTION 1 L --->!17         32!<--- MB 16 L
SEC FUNCTION 2 L --->!18         31!<--- MB 08 L
SEC FUNCTION 0 L --->!19         30!<--- MB 12 L
        SB 00 H <----!20         29!<--- MB 04 L
        SB 04 H <----!21         28!<--- MB 20 L
        SB 16 H <----!22         27!<--- MB 00 L
        SB 20 H <----!23         26!---> SB 32 H
        SB 24 H <----!24         25!---> SB 28 H
                              ---------------
```

SRM FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.26 TIMER (TOK)

1. **GENERAL DESCRIPTION:**
TOK is a 16 bit programmable timer. The initial count value is held in a register called INIR. The counter is a register called IICR. There is a control and status register called TCSR. All registers are accessed from the WBUS under the control of the inputs WCTRL <5:0> H.

TOK TABLE I

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	1 USEC CLK EN L		GA1TPF
2	WBUS 19 H		GA1TZF
3	WBUS 23 H		GA1TZF
4	WBUS 24 H		GA1TZF
5	TIMER SERVICE H		GA1TTF
6	WBUS 31 H		GA1TZF
7	WBUS 18 H		GA1TZF
8	WBUS 20 H		GA1TZF
9	WBUS 17 H		GA1TZF
10	PROC INIT L		GA1TNF
11	TIMER INT L		GA1TTH
14	WBUS 22 H		GA1TZF
15	WBUS 21 H		GA1TZF
16	WBUS 16 H		GA1TZF
17	B CLK L		GA1TNF
20	D CLK ENABLE H		GA1TNF
21	WCTRL 3 H		GA1TNF
22	WCTRL 1 H		GA1TNF
23	WCTRL 4 H		GA1TNH
24	WCTRL 5 H		GA1TNH
25	WCTRL 0 H		GA1TNF
26	WCTRL 2 H		GA1TNF
27	WBUS 00 H		GA1TZF
28	WBUS 03 H		GA1TZF
29	WBUS 01 H		GA1TZF
31	WBUS 02 H		GA1TZF
32	WBUS 04 H		GA1TZF
33	WBUS 05 H		GA1TZF
34	WBUS 06 H		GA1TZF
36	WBUS 07 H		GA1TZF
39	WBUS 09 H		GA1TZF
40	WBUS 08 H		GA1TZF
41	WBUS 10 H		GA1TZF
42	WBUS 11 H		GA1TZF
43	WBUS 13 H		GA1TZF
45	1 USEC CLK H		GA1TZF
46	WBUS 14 H		GA1TZF
47	WBUS 15 H		GA1TZF
48	WBUS 12 H		GA1TZF

2 Performance Requirements:

CLOCKS: Three clock signals are input to TOK:

1. B CLK L - Received Only.
2. D CLK ENABLE H - Received Only.
3. USEC CLK H - Tri-state Transceiver.
   If 1 USEC CLK EN H = Ø,
   then 1 USEC CLK H = REC CLK L
   if 1 USEC CLK EN H = 1,
   then 1 USEC CLK H is used
   only as an input (driver high impedence)

2.1 D CLK L: Another clock signal, D CLK L, is generated internally from B CLK L and D CLK ENABLE H, where:

```
D CLK L = B CLK L + D CLK ENABLE L.
```

2.2 REC CLK H, DLY CLK 1 H, DLY CLK 2 H: Three edge triggered D flipflops are used in generating INC CLK H, REC CLK H, DLY CLK 1 H, DLY CLK 2 H. These operate as a function of 1 USEC CLK EN L. Each of these are clocked on the rising edge of B CLK L.

If 1 USEC CLK EN H = 1, then

```
REC CLK H <- 1 USEC CLK H
DLY CLK 1 H <- REC CLK H
DLY CLK 2 H <- DLY CLK 1 H
```

If 1 USEC CLK EN H = Ø, then

REC CLK H, DLY CLK 1 H, DLY CLK 2 H form a divide by eight counter. REC CLK H is the least significant bit. DLY CLK 2 H is the most significant bit.

2.3 INC CLK H: INC CLK H is the output of an edge triggered D flipflop clocked on the rising edge of B CLK L.

```
INC CLK H <- SETTING SC L * SETTING TL
    * [SC H + TH + RUN H * REC CLK H *
      DLY CLK 1 H * DLY CLK 2 L]
```

where

```
SETTING SC L = .NOT. (WBUS 21 H * D CLK ENABLE H
    * (WCTRL <5:Ø> H = ØB))

SETTING T L = .NOT. (WBUS 2Ø H * D CLK ENABLE H
    * (WCTRL <5:Ø> H + ØB))
```

2.4 INIR <15:0> H: INIR is a 16 bit register. It is made of transparent latches.

To load INIR,

```
if D CLK H * (WCTRL <5:0> H = 0A) (ie. the latch is
open)
then INIR <15:0> H <- WBUS <15:0> H.
To read INIR.
if WCTRL <5:0> H = 0E,
then WBUS <15:0> H = INIR <15:0> H.
```

2.5 Timer Control and Status Register (TCSR): This register contains 10 bits. These bits include status information and control functions. When WCTRL <5:0> H = 0F, TCSR <31,24:16> is read out onto WBUS <31,24:16> H. When WCTRL <5:0> H = 0B, TCSR <31,24:16> is written from WBUS <31,24:16> H as described below.

```
           /|
+---+/ | /--+---+---+---+---+---+---+---+---+---+
|   |  |/   |   |   |   |   |   |   |   |   |   |
| E |       | T | I | I | S | T | S | T | V | R |
|   |       | V | R | E | C |   | R | R | P |   |
|   |       | P |   |   |   |   |   |   |   |   |
|   | /|    |   |   |   |   |   |   |   |   |   |
+---+/ | /--+---+---+---+---+---+---+---+---+---+
       |/
 31         24  23  22  21  20  19  18  17  16
```

BIT	NAME	FUNCTION
31	ERROR (E)	This bit is set when: (1)There is a carry out of IICR<15> and SR is set. (2)There is a carry out of IICR<15> with VP and IR set. It is cleared by writing a one to TCSR <31> with the low pulse of D CLK L.
24	TRANSFER OVERFLOW PENDING (TVP)	This bit is loaded from WBUS<24> H with the low pulse of D CLK L.
23	INTERRUPT REQUEST (IR)	This bit is set when there is a carry out of IICR<15> and VP is set. It is cleared by writing a one to TCSR <23> with the low pulse of D CLK L.
22	INTERRUPT ENABLE (IE)	This bit is loaded from WBUS <22> H, with the low pulse of D CLK L.

21 SINGLE CLOCK (SC) This bit is loaded from WBUS <21> H with the low pulse of D CLK L SC is asynchronously cleared with the high pulse of B CLK L if INC CLK H was set with the rising edge of that pulse.

20 TRANSFER (T) Writing a one to this bit will cause the IICR to be loaded with the value contained in the INIR. T is an edge triggered D flipflop. It is loaded from WBUS <20> H with the rising edge of D CLK L. It is cleared with the rising edge of B CLK L if INC CLK H.

19 SERVICE REQUEST (SR) This bit is set when there is a carry out of IICR<15> unless VP is set. It is cleared by writing a one to TCSR<19> with the low pulse of D CLK L.

18 TRANSFER REQUEST (TR) This bit is set when there is a carry out of IICR<15> and VP is set and TVP is clear. It is also set when T is being loaded with a one. It is cleared by writing a one to TCSR<18> with the low pulse of D CLK L.

17 OVERFLOW PENDING (VP) This bit is set by writing a one to TCSR<17> with the low pulse of D CLK L. It is not affected by writing a 0 to this bit. It is loaded from TVP with the low pulse of D CLK L when T is being loaded with a one. It is cleared when there is a carry out of IICR <15> if VP is set and TVP is clear.

16 RUN (R) This bit is loaded from WBUS<16> H with the low pulse of D CLK L.

2.6 IICR: IICR <15:0> is a 16 bit synchronius loadable counter. It is clocked with the rising edge of INC CLK H.

If T H + (IICR <15:0> = FFFF) * VP H,
then IICR <15:00> <- INIR <15:0>

otherwise, IICR <15:0> is incremented by 1.

If WCTRL <5:0> H = 0F, IICR <15:0> is read out onto WBUS <15:0> H.

2.7 **TIMER SERVICE H:** TIMER SERVICE H is the output of an edge triggered D flipflop clocked on the rising edge of D CLK L.

TIMER SERVICE H <- SR H + TR H.

2.8 **TIMER INT L:** TIMER INT L is the low true output of an edge triggered D flipflop, clocked on the rising edge of D CLK L.

TIMER INT H <- IR H * IE H.

2.9 **PROC INIT L:** If PROC INIT L is low, all bits of the TCSR and the TIMER INT flipflop are asynchronously cleared.

```
                    ---------------
1 USEC CLK EN L <-->!01          48!<--> WBUS 12 H
      WBUS 19 H <-->!02          47!<--> WBUS 15 H
      WBUS 23 H <-->!03          46!<--> WBUS 14 H
      WBUS 24 H <-->!04          45!<--> 1 USEC CLK H
TIMER SERVICE H <-->!05          44!
      WBUS 31 H <-->!06          43!<--> WBUS 13 H
      WBUS 18 H <-->!07          42!<--> WBUS 11 H
      WBUS 20 H <-->!08          41!<--> WBUS 10 H
      WBUS 17 H <-->!09          40!<--> WBUS 08 H
    PROC INIT L --->!10 ....... 39!<--> WBUS 09 H
    TIMER INT L <-->!11 .     . 38!---- GROUND
            VGA ----!12 . LID . 37!
            VCC ----!13 . DOWN. 36!<--> WBUS 07 H
      WBUS 22 H <-->!14 .     . 35!---- GROUND
      WBUS 21 H <-->!15 ....... 34!<--> WBUS 06 H
      WBUS 16 H <-->!16          33!<--> WBUS 05 H
        B CLK L --->!17          32!<--> WBUS 04 H
                    !18          31!<--> WBUS 02 H
                    !19          30!
 D CLK ENABLE H --->!20          29!<--> WBUS 01 H
      WCTRL 3 H --->!21          28!<--> WBUS 03 H
      WCTRL 1 H --->!22          27!<--> WBUS 00 H
      WCTRL 4 H --->!23          26!<--- WCTRL 2 H
      WCTRL 5 H --->!24          25!<--- WCTRL 0 H
                    ---------------
```

TOK FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.27 UNIBUS CONTROL (UCN) DC 619

1. **GENERAL DESCRIPTION:**

This specification defines the detail requirements for the VAX-11/750 Unibus Adaptor.

2. **APPLICABLE DOCUMENTS (Per latest revision on date of order):**

Digital Equipment Corporation

A-PS-1900003-GS	General Purchase Specification for A 400 Internal Gate, Low-Power Schottky Type, Gate Array Integrated Circuit DC 600 Series

UCN TABLE 1

Pin Identification and Gate Type

PIN	PIN		GATE	PIN	PIN		GATE
NO.	ID	COMMENTS	TYPE	NO.	ID	COMMENTS	TYPE
1	BUF CMI 31 H		GA1TZF	25	OFFSET H		GA1TZF
2	MSYN H		GA1TNF	26	SCØ H		GA1TTN
3	CSRA 4 H		GA1TNG	27	DPSEL Ø H		GA1TZF
4	CSRA 11 H		GA1TNG	28	DPSEL 1 H		GA1TZF
5	ADDU H		GA1TNF	29	MAP CTRL EN L		GA1TTN
6	ADDC H		GA1TNH	3Ø	CSRA 2 H		GA1TNG
7	BLOCK ARB L		GA1TTN	31	B CLK L		GA1TNF
8	C1 H		GA1TZF	32	CSRA 3 H		GA1TNG
9	HIGHEST L		GA1TNF	33	BUT Ø H		GA1TNF
1Ø	URA 2 L	I_{oL}=2Øma	GA1TCN	34	BUF CMI 25 H		GA1TZF
11	MATCH H		GA1TZF	36	INT H		GA1TNH
14	BUF CMI 28 H		GA1TZF	37	URA 1 L	I_{oL}=2Øma	GA1TCN
15	CØ H		GA1TZF	39	BUF CMI 3Ø H		GA1TZF
16	URA 3 L	I_{oL}=2Øma	GA1TCN	4Ø	BUF CMI 29 H		GA1TZF
17	AØ H		GA1TZF	41	BUF CMI Ø H		GA1TZF
18	A1 H		GA1TZF	42	INIT H		GA1TNF
19	BUT 1 H		GA1TNF	43	SSYN H		GA1TNF
2Ø	BUT 2 H		GA1TNF	44	STATUS 1L	I_{oL}=2Øma	GA1TCF
21	URA Ø L	I_{oL}=2Øma	GA1TCN	45	STATUS ØL	I_{oL}=2Øma	GA1TCF
22	BUF CMI 27 H		GA1TZF	46	DBBZ L	I_{oL}=2Øma	GA1TCF
23	SC1 H		GA1TTN	47	TIMOUT H		GA1TNF
24	BUF CMI 26 H		GA1TZF	48	PH H		GA1TZF

2. Performance:

2.1 Conventions:

This functional description makes use of internal state elements and internal signals to simplify some of the details. All internal signal names are noted by the fact that they end with a period. An index is provided at the end of the section to assist in finding the definition.

The & is used to represent a logical "AND", the + for logical "OR". All equations assume positive logic, a logical one is high. Signal names without a postfixed assertion letter are high true. An "L" after a signal name indicates a low true assertion. If an input pin is labeled as NAME L, then the following holds true:

Level on input pin	Name	Name L
L	H	L
H	L	H

Certain output pins are low true asserted. For some of these the description actually defines the complement of the pin.

Most state elements are clocked based on the input pin BCLK L. Unless stated otherwise, edge triggered devices change state on the rising edge of BCLK L, and set-reset devices have the set and clear conditions ANDed with BCLK H.

Certain input pins are defined as having "Skew Latches". These are feed through latches that are open when BCLK L is high and closed when BCLK L goes low.

2.2 Internal State:

2.2.1 BYTE FLAGS:

UCN contains 15 SET-RESET flipflops called byte flags. These are organized as three groups of 5 bits per group. The groups are associated with the three buffered data paths and a group is selected by the value cf DPSEL 1 H and DPSEL Ø H. The groups are numbered 1 through 3. When these two lines have the value ØØ then no state is changed within these bits. The outputs are selected by a multiplexer controlled by the DPSEL lines. When these lines have the value ØØ the output of the multiplexer has the value Ø.

The five bits are labelled BF3, BF2, BF1, BFØ, and CD. The outputs of the multiplexer will be referred to as

BF3., BF2., etc. A set of signals known as BFmX. is used throughout this spec. They are defined as follows.

if: (BUT210 ≠ HHH) & (PURGE EVENT L. & C1 L & INT L)
_7 then BFmX. = 1

else if:
DP0. = 1

then BFmX. = function of A1, A0, C1, C0, offset such that if Table 1 has an *, BFmX. = 1, otherwise 0. For this case BF0X. will also equal 1 if BUT210 = LHH.
else:
BFmX. = BFn.

When BUT2, BUT1, BUT0 = HLH, then CD is cleared and the BF bits are set as a function of A1, A0, C0, and OFFSET as described by Section 2.5.1.

When BUT2, BUT1, BUT0 = LHH, then BF0 is set.

When STATUS TIME. = H, then BF3 - BF0 are cleared and CD is cleared if C1 L = L or PURGE SEL 1. = H or PURGE SEL 0. = H. If it is not cleared, then it is set.

INIT H = H clears all bits independent of the state of BCLK or DPSEL.

Whenever a PURGE REQ flipflop is cleared (see Section 1.12), the corresponding CD is cleared.

STATUS TIME. is defined in Section 1.10.

2.2.2 ERROR BITS - UCE, NXM:

The error bits consist of 3 groups of 2 bits (SET/RESET flipflop) each. The selection of the groups for both clocking and readout is controlled by the DPSEL lines or the CSRA3, CSRA2 lines. The groups are numbered 1 through 3 and are selected by the equivalent values on the DPSEL or CSRA lines. When the appropriate lines have the value 00 then no change in state occurs on setting/clearing and the outputs of the selection MUX have the value of 0.

NXM is set when STATUS TIME. = H and STATUS 1, STATUS 0 = LL. Selection of group is controlled by DPSEL.

NXM is cleared when CLK CSR B3. = H and BUF CMI 30 = H. Selection is controlled by CSRA 3,2.

UCE is set when STATUS TIME. = H and STATUS 1, 0 = LH. DPSEL selects.

UCE is cleared when CLK CSR B3. = H and BUF CMI 29 = H. CSRA 3,2 selects.

All six bits are cleared when INIT = H, independent of BCLK.

CLK CSR B3. = CSRA 11 L & CSRA 4 L & SST 1 H. & SST 0 L. & LATCH CMI 31 H.

There are three outputs from the two bits.

DP NXM. = NXM selected by DPSEL

REG NXM. = NXM selected by CSRA 3,2

REG UCE. = UCE selected by CSRA 3,2.

2.2.3 LATCH CMI <31:25>:

These are 7 latches whose inputs are BUF CMI <31:25>. The latches are open when DBBZ L = L, and PREV DBBZ. = L.

2.2.4 DATIP SEQ H:

This is a SET-RESET flipflop. It is set when C1 = L, C0 = H, DPSEL 1,0 = 00, BUT 2,1,0 = HHH, and MSYN = H.

It is cleared when C1 = H, STATUS TIME. = H, PURGE SEL. 1,0 = 00. It is also cleared when INIT = H, independent of BCLK.

2.2.5 SST. 2,1,0:

These three edge triggered D flipflops form a sequential machine that control various functions inside the device. SST2. is cleared by INIT independent of BCLK.

Definition of D inputs:

SST 2: CSR ADDR. & BUF CMI 27 L + CSRA 11 & CSRA 4 & SST2. & SST1.

SST 1: CSR ADDR. & (CSRA 11 + CSRA 4 L) & SST 2. & SST 1.

SST 0: CSR ADDR. & BUF CMI 27 L + SST 2.

CSR ADDR. = ADDC & DBBZ H & PREV DBBZ L. & (BUF CMI 27 L + BUF CMI 26 L)

2.2.6 LATCH ADDU:

LATCH ADDU is an edge triggered D flipflop. Its D input is:

ADDU & DBBZ H & PREV DBBZ L. & (BUF CMI 27 L + BUF CMI 26 L) + LATCH ADDU. & SSYN L & TIM OUT L

It is cleared by INIT = H independent of BCLK.

2.2.7 DRIVE UBUS CTRL:

This is an edge triggered D flipflop. Its D input is:

LATCH ADDU. + DRIVE UBUS CTRL. & MSYN H.

2.2.8 DRIVE PB:

This is a SET-RESET flipflop. It is set when STATUS TIME. = H, STATUS 1,Ø = LH. It is cleared by MSYN = L, independent of BCLK.

2.2.9 PREV DBBZ:

This is an edge triggered D flipflop whose D input is DBBZ.

2.2.1Ø CMI CYCLE:

This is an edge triggered D flipflop. Its D input is:
(BUT 2,1,Ø = LHX & HIGHEST & DBBZ L + CMI CYCLE. & DBBZ H & INIT L.
STATUS TIME. = CMI CYCLE. & DBBZ L.

2.2.11 CMI LOCKED:

This is an edge triggered D flipflop. Its D input is:

(BUF CMI 27 L & ADDR NO WV. & CMI 25 H +

CMI LOCKED & INIT L & (ADDR NO WV. & BUF CMI 27 & BUF CMI 25)

ADDR NO WV. = DBBZ H & PREV DBBZ L & (BUF CMI 27 L + BUF CMI 26 L).

2.2.12 CSR PURGE REQ 3,2,1:

These three SET-RESET flipflops are selected to be set and cleared by CSRA 3,2 and DPSEL 1,Ø, respectively.

A given one is set when selected by CSRA3,2 and CSRA11 = L, CSRA4 = L, SST1. = H, SSTØ. = L, LATCH CMI 28. = H, BUF CMI Ø = H.

It is cleared when selectec by DPSEL and the following condition is true:

(BUT 210 = LLH & NOT EMPTY. L) + PURGE EVENT. & CMI CYCLE. & DBBZ L.

NOT EMPTY. = BF3X. + BF2X. + BF1X. + BF0X.

PURGE EVENT. is defined in Section 2.2.13.

All three bits are cleared by INIT = H independent of BCLK.

CSR PURGE REQ. = CSR PURGE REQ2. + CSR PURGE REQ1. + CSR PURGE REQ 0.

2.2.13 PURGE SEL 1,0:

These two bits do not follow the clocking conventions described in the beginning of this section. They are set when the set conditions below are met and BCLK L is low. They are cleared on the rising edge or BCLK L when the clear conditions are met.

PURGE SEL 1.

SET: DPSEL 1 & SET PURGE. + (PURGE REQ2. + PURGE REQ3.) & (BUT2,1,0 = HHH).

CLEAR: STATUS TIME. + (BUT2,1,0 = LLH) & NOT EMPTY L.

PURGE SEL 0.

SET: DPSEL 0 & SET PURGE.+(PURGE REQ3.+(PURGE REQ2 L. & PURGE REQ1.) & (BUT 2,1,0 = HHH).

CLEAR: SAME AS PURGE SEL 1.

SET PURGE. = (BUT 2,1,0 = HHH) & MSYN & DP NXM L & CSR PURGE REQ L & NOT EMPTY. & DP0 L. & (C1 L + MATCH L + C1 & C0 & A1 & A0 & OFFSET).

PURGF EVENT. = PURGE SEL 1. + PURGE SEL 0.

2.3 Input Only Pins:

Input pins will only be described in general terms in this section. The precise use of inputs will be defined throughout the spec when they are used to define outputs or internal state elements.

2.3.1 ADDC H:

This line is used to indicate that an address on the CMI is in Unibus CSR space.

2.3.2 ADDU H:

This line is used to indicate an address on the CMI that is in Unibus Address space.

2.3.3 BUT H (BUT2, BUT1, BUT0):

These lines control the clocking of the byte flags, purge SEL registers, and CMI cycle flip flop. They also control the URA output lines.

2.3.4 CSRA 2, 3, 4, 11 H:

These lines correspond to CMI address lines of the same number. They are used to select between the Map and CSR's, and also to select which CSR.

CSR11	CSR4	CSR3	CSR2	
1	X	X	X	MAP
0	0	X	X	CSR'S
0	1	X	X	RESERVED

2.3.5 HIGHEST L:

This line indicates that the UBI is the highest priority device arbitrating for the CMI.

2.3.6 INIT H:

This line initializes most state elements within the device.

2.3.7 INT H:

This line comes from a synchronized version of Unibus INTR.

2.3.8 MSYN H:

A synchronized version of Unibus MSYN.

2.3.9 SSYN H:

A synchronized version of Unibus SSYN.

2.3.10 TIMOUT H:

This line indicates that the UBI timed out waiting for a SSYN on the Unibus.

2.4 Output Only Pins:

2.4.1 BLOCK ARB L:

BLOCK ARB L = CMI LOCKED. & DP0. & C1 L& C0 & BUT 0 L

2.4.2 MAP CTRL EN L:

MAP CTRL EN L = SST1 L.& SST0. + PURGE EVENT.& SST0 L.
+ INT

2.4.3 SC1, SC0:

SC1 H = SST1 L.
SC0 H = SST0.

2.4.4. URA (3, 2, 1, 0) L. OPEN COLLECTOR:

These lines are used to control the UBI micro-code. They are intended to be wire-ored with the micro-code address lines. They are controlled by the BUT lines plus other signals as defined in UCN Tables 3 and 4.

2.5 Bidirectional Pins:

2.5.1 A0, A1, C0, C1:

These four pins are all tri-state. The tri-state enable is:

DRIVE UBUS CTRL.

```
A1 =  LATCH CMI 29 L. & LATCH CMI 28 L.
A0 =  LATCH CMI 30 L. & LATCH CMI 28 L. & LATCH CMI 27 H.
C0 =  LATCH CMI 27 H.
C1 =  A0. + LATCH CMI 25. & LATCH CMI 27 L. +
      LATCH CMI 27 L. & LATCH CMI 26 H. +
      LATCH CMI 31 L. & LATCH CMI 29 L. & LATCH CMI 27 H.
```

2.5.2 BUF CMI <31:25, 0>:

These 8 lines are tri-state. The tri-state enable is:

```
SST1 L. & SST0 H. + SST0 H. & CSRA11 L +
CMI CYCLE.& PREV DBBZ L. +
(BUT 210 = LHX) & (PREV DBBZ. + DBBZ L)
BUF CMI 31 H = BF3X. + SST1 & (REG NXM + REG UCE)
BUF CMI 30 H = BF2X. + SST1 & REG NXM
BUF CMI 29 H = BF1X. + SST1 & REG UCE
BUF CMI 28 H = BF0X.
BUF CMI 27 H = (PURGE EVENT. + C1 L + INT) & SST 0 L.
BUF CMI 26 H = PURGE EVENT L. & INT & SST0 L.
BUF CMI 25 H = DATIP SEQ. & PURGE EVENT L. & SST0 L.
BUF CMI 0 H = PURGE REQ.
```

2.5.3 DBBZ L:

This is an open collector pin. DBBZ is received through a skew latch and it is this ʟatched DBBZ that is actually used thoughout this specifɪcation. On output:

```
DBBZ H =   LATCH ADDU. + SST2. +
           CMI CYCLE. & PREV DBBZ L.
```

2.5.4 DPSEL 1,0 H:

These two lines are tri-state. The tri-state enable is: SST1 L. & SST0. + PURGE EVENT.& SST0 L. + INT

DPSEL 1 = CSRA3 & PURGE EVENT L. + PURGE SEL1.
DPSEL 0 = CSRA2 & PURGE EVENT L. + PURGE SEL0.

2.5.5 MATCH H:

This is a tri-state pin. The enable is: BUT210 = LHX.
MATCH H = HIGHEST L.

2.5.6 6OFFSET H:

This is an open collector pin.

OFFSET H = INT L.

2.5.7 PB:

This is a tri-state pin. The enable is:
C1 L & MSYN & DRIVE UBUS CTRL L.

PB H = DRIVE PB.

2.5.8 STATUS 1,0 L:

These are two open collector pins. They are received through skew latches and' it is the output of these latches that are used throughout this specification.

STATUS 1 H = SST1. + SST0. + PB L & DRIVE UBUS CTRL. & SSYN & MSYN
STATUS 0 H = SST1 + SST0. + DRIVE UBUS CTRL. & SSYN & MSYN.

UCN TABLE 2

SETTING OF BYTE FLAGS

C0	A1	A0	OFFSET	BF3	BF2	BF1	BF0
L	L	X	L			*	*
L	L	X	H		*	*	
L	H	X	L	*	*		
L	H	X	H	*			
H	L	L	L				*
H	L	L	H			*	
H	L	H	L			*	
H	L	H	H		*		
H	H	L	L		*		
H	H	L	H	*			
H	H	H	L	*			
H	H	H	H				

* Means bit is set, blank means no change.

UCN TABLE 3

URA OUTPUTS

BUT 2	BUT 1	BUT 0	URA 3 L	URA 2 L	URA 1 L	URA 0 L
L	L	L	H	H	H	H
L	L	H	H	H	MSYN L	EMPTY PURGE L.
L	H	L	H	H	MSYN L	HIGHEST & DBBZ L
L	H	H	H	H	MSYN L	HIGHEST & DBBZ L
H	L	L	H	H	MSYN L	TIMOUT L & SSYN L
H	L	H	H	H	MSYN L	TIMOUT L & SSYN L
H	H	L	H	WRAP.	DBBZ	STATUS 1L & STATUS 0L
H	H	H	----------SEE UCN TABLE 4--------------------------			

WRAP. = A1 & OFFSET & (C1 L + C0 L)

UCM TABLE 4

URA OUTPUTS FOR BUT21Ø = HHH

```
URA 3 L =  DPØL.& FF UBUS. & AUTO PURGE L.

URA 2 L =  DPØL.& FF UBUS. & DATOB WRAP L. +
           LATCH ADDU. +
           FF UBUS. & C1 & AUTO PURGE L. & DATOB WRAP L.

URA 1 L =  LATCH ADDU +
           DPØL.& C1 L. & CØ& MATCH +
           DPØL.& C1 H & FULL.& DATOB WRAP L. +
           PURGE.

URA Ø L =  DPØ.& FF UBUS.& C1 +
           LATCH ADDU. & LATCH CMI 27. +
           DPØL.& C1 L.& WRAP. & CØ& FF UBUS. +
           INT & PURGE EVENT L.

DPØ. =     DPSEL 1 L & DPSEL Ø L

FF UBUS. = MSYN & SST 1 L. & DP NXM L. & CSR PURGE REQ L.

AUTO PURGE. =   NOT EMPTY. & DPØ L. & (C1 L + MATCH L. + DATOB WRAP.)

DATOB WRAP. =   C1 & CØ & A1 & AØ & OFFSET

WRAP. =    A1 & OFFSET & (C1L + CØL)

FULL. =    (BF3X. + BF3 WILL GET SET PER TABLE I)
           (BF2X. + BF2 WILL GET SET............)
           (BF2X. + ...)
           (BFØX. + ...)
           + WRAP.
```

```
                         ---------------
BUF CMI 31 H <--->!01            48!<--> PB H
       MSYN H --->!02            47!<--- TIMOUT H
     CSRA 4 H --->!03            16!<--> DBBZ L
    CSRA 11 H --->!04            45!<--> STATUS 0 L
       ADDU H --->!05            44!<--> STATUS 1 L
       ADDC H --->!06            43!<--- SSYN H
 BLOCK ARB L <---!07             42!<--- INIT H
         C1 H <-->!08            41!<--> BUF CMI 0 H
   HIGHEST L --->!09             40!<--> BUF CMI 29 H
     URA 2 L <---!10 ........ 39!<--> BUF CMI 30 H
     MATCH H <-->!11 .        . 38!---- GROUND
         VGA ----!12 . LID . 37!---> URA 1 L
         VCC ----!13 . DOWN. 36!<--- INT H
BUF CMI 28 H <-->!14 .        . 35!---- GROUND
        C0 H <-->!15 ........ 34!<--> BUF CMI 25 H
     URA 3 L <---!16             33!<--- BUT 0 H
        A0 H <-->!17             32!<--- CSRA 3 H
        A1 H <-->!18             31!<--- B CLK L
     BUT 1 H --->!19             30!<--- CSRA 2 H
     BUT 2 H --->!20             29!---> MAP CTRL EN L
     URA 0 L <---!21             28!<--> DPSEL 1 H
BUF CMI 27 H <-->!22             27!<--> DPSEL 0 H
       SC1 H <---!23             26!---> SC0 H
BUF CMI 26 H <-->!24             25!<--> OFFSET H
                         ---------------
```

UCN FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION

3.28 UNIBUS DATA PATH (UDP)

1. Introduction

1.1 The UDP chip is used in Unibus Interface logic (UI). It connects the Unibus Data and Address lines to the CMI Addιess/Data lines. It contains three sets of registers to buffer two byte Unibus transactions to four byte CMI transactions.

The chip is designed as a data slice containing two bits per byte of data and four consecutive address bits.

1.2 General Description

The three Buffered Data Path Registers (BDP) can be loaded from either the Unibus or CMI.

They can be sourced onto either the Unibus or CMI. Data is stored aligned to the CMI. It is rotated going to or from the Unibus.

The device has four ports. The UD port is used to input data from the Unibus to the device and also to output data from the device to the Unibus. It is a bi-directional tri state port and it is intended to be used with a Unibus to tri state transceiver.

The UA port is used to receive Unibus Addresses for storage and comparison. It also is used to output stored addresses as well as outputing CMI supplied addresses to be used within the CUI (not going to the Unibus). The UA is a bidirectional tri state port.

The CMI port is a bidirectional tri state port that ties directly to the CMI for rdriving and receiving addresses and data.

The BUF CMI port is a bidirectional tri state port through which received addresses or data can be made available to the rest of the CUI. the CUI can also supply addresses and data to be driven onto the CMI.

The above features are primarily controlled by three groups of control lines. The BDPC lines control the loading of the BDP's and BAR's. The PRTC lines control the steering of the ports and the SC lines modify the intent of the PRTC's for port control.

All register clocking is done with a suplied clock, BCLK L. A cycle is considered to be from one rising edge of BCLK L to the next. The chip generally expects that all control inputs can change just after the rising edge of BCLK L and will be stable by the time it falls.

Internal state changes are usually made with the low portion of BCLK L and control is done interrnally by gating off this low pulse. Unless otherwise stated, all registers are feed througyh latches that are open when BCLK L is low.

2.0 Registers

2.1 CMI Latch

This is an 8 bit feedthrough latch on the incoming CMI lines. It is open when BCLK L is high and closes when BCLK L goes low. It is designed to enable latches inside the chip to meet their hold time requirements even though the CMI timing specification allows data to disappear 2ns before the rising edge of BCLK. All references to the CMI for the remainder of this specification actually refer to the output of this latch, not the CMI pins on the chip.

2.2 BDP Registers

UDP Table 1 indicates the loading of data into the BDP registers. There are three registers and the particular one to get loaded is selected by the value of the input pins DP SEL. Values of 1, 2, or 3 select one of the three registers. None are loaded for a value of 0 (see section 4.5). Each register has two bits per byte. Both are treated identically, therefore the chart shows how the registers are loaded on a per byte basis.

2.3 UD Latches

These are feedthrough latches that drive the UD pins. Their operation is described in Chart. The latches are actually constructed by tying the receiver output to one of the driver inputs. It is therefore necessary for the tri state enable of the UD pins to be enabled in order for the latch to close.

2.4 Buffered Address Registers (BAR)

There are three BAR's, corresponding to the non zero values of DP SEL. A BAR is clocked when BDPC2 is high. They are four bit latches, each bit is loaded from a UA pin.

2.5 Received CMI Address Register (RCAR)

The RCAR is a three (0, 1, 2) byte, two bits per byte register. It is loaded from the CMI and is clocked on any cycle for which DBBZ L is low and previous DBBZ H is low.

3.0 State Control Elements

3.1 Drive UA

Drive UA is the output of an S-R FF whose direct clear input is the "or" of SC0 H, SCI L and PRTC = LHH. It is directly set by the "and" of SC0 L, SCI H, BCLK L, and the output of an S-R FF that is set by BCLK L being low and cleared by PRTC210 = LHH. On both FF's simultaneous assertion of set and clear causes the output to be high.

3.2 Drive CMI

Drive CMI is a two bit register that controls the enabling of the CMI tri state drivers and also the source of the CMI data. It is clocked on the rising edge of BCLK. The sources of CMI drive and functions that enable the drivers are described in section 4.3.

4.0 Pin List

4.1 UD

These four lines connect to the Unibus through a Unibus to tri state transceiver. The full names for the lines are:

UD 00	Byte zero, first bit
UD 01	Byte zero, second bit
UD 08	Byte one, first bit
UD 09	Byte one, second bit

Both bits within each byte are handled the same way. UDP Table 2 shows what is output on these lines. These lines are actually the output of the UD latch. The output is a function of the PRTC lines (section 4.9), the A1 line (4.6), and the offset line (4.7). these lines are asserted high.

4.2 UA

These four lines receive the Unibus Address. They can be sourced from the BARs or the RCAR. The output is controlled by the PRTC lines (4.9), the BUF SEL, and the SC lines (4.10). UDP Table 3 lists the ouput conditions for these lines. The full names for tha four lines are:

UA 0
UA 1
UA 2
UA 3

These lines are asserted high.

4.3 CMI

These eight lines tie directly to the CMI. There are two bits per byte. The full names are:

CMI 00	Byte 0 first bit
CMI 01	Byte 0 second bit
CMI 08	Byte 1 first bit
CMI 09	Byte 1 second bit
CMI 16	Byte 2 first bit
CMI 17	Byte 2 second bit
CMI 24	Byte 3 first bit
CMI 25	Byte 3 second bit

UDP Tables 4 and 5 describe the sources of information driven onto these lines. The tri state enable for these lines and the data on them is actually enabled by the rising edge of BCLK that occurs after the PRTC or SC lines have been enabled. See section 3.2. These lines are asserted high.

4.4 BUF CMI

These are 8 bidirectional tri state lines are used to provide addresses, map entries, and CSR data to the CMI when used as inputs. When used as outputs they provide CMI addresses to the Unibus and data to write the map and CSR's within the CUI.

They are primarily under the control of the PRTC lines and the SC lines as shown in UDP Table 6. Whenever the PRTC210 lines have the value LHX, PREV DBBZ has the value L, and DBBZ L has the value L, then Byte 3 of the BUF CMI has the value of CMI B3 and not Hi-Z as shown in the chart. These lines are asserted high.

4.5 DP SEL

These are two input lines that select which of four Data Paths to use for values of 1, 2, 3, it selects data from one of the BDP's. For values of 0, the chip typically behaves in a different manner. These differences are called out in the appropriate sections. Full names are DP SEL 0 and DP SEL 1. These lines are asserted high.

4.6 A1, A0

These input lines come from the Unibus A1, A0 lines and are generally used to specify whether the Unibus data goes to/comes from the upper or lower two bytes of the various four byte ports and registers. These lines are asserted high. The A0 line is only used when the BDPC lines indicate a DATOB function.

4.7 OFFSET

The input is the byte offset line which comes from the Map. It specifies that the Unibus data should be treated as though its address is one byte higher than the Unibus Address. This line is asserted high.

4.8 BDPC

These 3 lines specify how the BDP and BAR latches should be loaded. Their use is precisely specified in Section 2. UDP Table 7 summarizes the control functions.

4.9 PRTC

These 3 inputs control the steering fo the ports. They are described in detail in sections 4.1, 4.2, 4.3, and 4.4. UDP Table 8 summarizes how the ports are controlled for SC values of LL. Section 4.7 explains the affect of SC.

4.10 SC

These lines modify the activity of the chip's ports to be different from what is normally specified by the PRTC lines. Their intended use is to enable Reads and Writes to registers within the CUI itself. These include the BAR's (Read Only), the MAP, and the CSR's. The precise affect on the ports is described under each port. UDP Table 9 gives an overview of the effect of these lines.

4.11 MATCH

This pin is a bidirectional control signal. It is an open collector output.

When PRTC1 = H, this pin is an input. The output driver is disabled. It is used to tell the UDP that on the next rising edge of BCLK, it may assert its CMI drivers. See sections 4.3 for this function, low implies assert on CMI.

For all other values of PRTC, this pin is an output. High means that the value on the UA lines matches what is stored in the BAR selected by the DP SEL lines. Low means no MATCH.

4.12 BCLK L

BCLK is used to clock all of the state elements within the chip. The nature of the state element (edge triggered, transparent, master-slave, etc.) is defined along with its functional definition.

4.13 DBBZ L

DBBZ is a buffered copy of the DBBZ L signal on the CMI. Its use is defined in sections 2.5, 3.2, 4.3, and 4.4.

4.14 PREV DBBZ H

This pin is intended to receive a copy of CMI DBBZ that is clocked into an edge triggered D flop on the rising edge of BCLK L. Its use is detailed in sections 2.5 and 4.4.

4.15 ID1

ID1 is used to distinguish different UDP chips among the four that are used in the system. It is set as follows:

Bits/Byte	Chip #	ID1
0,1	0	L
2,3	1	H
4,5	2	L
6,7	3	L

4.16 ADDU, ADDC

These two lines are used to provide the CUI control logic with information as to whether or not the address on the CMI is either in the Unibus space or the CUI MAP and CSR space. These are open collector outputs.

The decode of the address lines changes with the ID pin.

ADDU = (CMI 17) (CMI 16)

ADDC = (CMI 17) (CMI 16) (CMI 09) (CMI08) for ID1 = L

ADDC = (CMI 17) (CMI 16) for ID1 = H

5.Ø Pin List Summary

PIN #	ELECTRICAL CHARACTERISTICS		SIGNAL NAME
	Transceiver	Tri-State	UD ØØ H
	Transceiver	Tri-State	UD Ø1 H
	Transceiver	Tri-State	UD Ø8 H
	Transceiver	Tri-State	UD Ø9 H
	Transceiver	Tri-State	UA Ø H
	Transceiver	Tri-State	UA 1 H
	Transceiver	Tri-State	UA 2 H
	Transceiver	Tri-State	UA 3 H
	Transceiver	Tri-State	CMI ØØ H
	Transceiver	Tri-State	CMI Ø1 H
	Transceiver	Tri-State	CMI Ø8 H
	Transceiver	Tri-State	CMI Ø9 H
	Transceiver	Tri-State	CMI 16 H
	Transceiver	Tri-State	CMI 17 H
	Transceiver	Tri-State	CMI 24 H
	Transceiver	Tri-State	CMI 25 H
	Transceiver	Tri-State	BUF CMI ØØ H
	Transceiver	Tri-State	BUF CMI Ø1 H
	Transceiver	Tri-State	BUF CMI Ø8 H
	Transceiver	Tri-State	BUF CMI Ø9 H
	Transceiver	Tri-State	BUF CMI 16 H
	Transceiver	Tri-State	BUF CMI 17 H
	Transceiver	Tri-State	BUF CMI 24 H
	Transceiver	Tri-State	BUF CMI 24 H
	Input		A1 H
	Input		AØ H
	Input		OFFSET H
	Input		BDPC 2 H
	Input		BDPC 1 H
	Input		BDPC Ø H
	Input		PRTC 2 H
	Input		PRTC 1 H
	Input		PRTC Ø H
	Input		SC 1 H
	Input		SC Ø H
	Transceiver	Open Collector	MATCH H
	Input		BCLK L
	Input		DBBZ L
	Input		PREV DBBZ H
	Input		1D1 H
	Output	Open Collector	ADDU H
	Output	Open Collector	ADDC H

UDP Table 1

BDP Loading

CHART ONLY APPLIES WHEN DP SEL 1.Ø = LH, HL, HH

CONTROL INPUTS						REGISTERS				NOTES
BDPC			A1	AØ	OFFSET	BDP B3	PDB B2	BDP B1	BDP BØ	
2	1	Ø								
L	L	L	X	X	X	*	*	*	*	1
L	L	H	X	X	X	CMI B3	CMI B2	CMI B1	CMI B1	
L	H	L	X	X	X					DON'T USE
L	H	H	X	X	H	*	*	*	*	
H	L	L	X	X	H	*	*	*	UDB1	
H	L	H	L	X	L	*	*	UD B1	UD BØ	
H	L	H	H	X	L	UD B1	UD BØ	*	*	
H	L	H	L	X	H	*	UD B1	UD BØ	*	
H	L	H	H	X	H	UD BØ	*	*	*	
H	H	L	L	L	L	*	*	*	UD BØ	
H	H	L	H	L	L	*	UD BØ	*	*	
H	H	L	L	L	H	*	*	UD BØ	*	
H	H	L	H	L	H	UD BØ	*	*	*	
H	H	L	L	H	L	*	*	UD B1	*	
H	H	L	H	H	L	UD B1	*	*	*	
H	H	L	L	H	H	*	UD B1	*	*	
H	H	L	H	H	H	*	*	*	UD B1	
H	H	H								DON'T USE

NOTES:

1. * Means register not loaded

UDP Table 2

UD LATCH SOURCES

PRTC	BDPC	A1	OFFSET	UD B1	UD B0	NOTES
LLH						1
LLH	LLH	L	L	SEL DP B1	SEL DP B0	2
LLH	LLH	H	L	SEL DP B3	SEL DP B2	
LLH	LLH	L	H	SEL DP B2	SEL DP B1	
LLH	LLH	H	H	SEL DP B0	*	
LLH	LHH	L	L	SEL DP B1	SEL DP B0	
LLH	LHH	H	L	SEL DP B3	SEL DP B2	
LLH	LHH	L	H	SEL DP B2	SEL DP B1	
LLH	LHH	H	H	SEL DP B0	SEL DP B3	
HLH	X	L	X	CMI B1	CMI B0	3
HLH	X	H	X	CMI B3	CMI B2	

1. When PRTC = LLH is first selected, the UD latch data is undefined. It will remain undefined until the first occurrence of BCLK L being low with BDPC = LHX or LLH. If PRTC is set to any value other than LLH following the latching of data, the latched data will be lost.

2. SELDP is: BDP if DPSEL = 1, 2, 3
   CMI latch if DPSEL = 0

3. UD is undefined until the first occurrence of BCLK L going L.

* Means not clocked.

UDP Table 3

UA SOURCES

PRTC			SC1	SCØ	UA3	2	1	Ø	NOTES
2	1	Ø							
			L	L					
L	L	X	H	L	H1-Z	H1-Z	H1-Z	H1-Z	
L	H	L	H	L	H1-Z	H1-Z	H1-Z	H1-Z	
L	H	H	H	L	SEL BAR 3	2	1	Ø	1
H	X	X	H	L	H1-Z	H1-Z	H1-Z	H1-Z	
X	X	X	H	H	SEL BAR 3	2	1	Ø	
X	X	X	L	X	RCAR Ø9	Ø8	Ø1	ØØ	

NOTES:

1. SEL BAR refers to the BAR selected by the value of DP SEL.

Table 4

CMI DRIVE

PRTC			SC	SC	MATCH	DBBZ L	CMI	NOTES
2	1	0	1	0				
L	L	L	H	L	X	X	HI-Z	
L	L	L	H	H	X	X	BUF CMI	1
L	L	H	H	L	X	X	HI-Z	
L	L	H	H	H	X	X	BUF CMI	
L	H	X	H	L	H	X	HI-Z	
L	H	X	H	L	L	H	BUF CMI	1
L	H	X	H	H	X	X	BUF CMI	
L	H	X	H	H	X	X	BUF CMI	
H	L	L	H	L	X	L	BDP	4
H	L	L	H	L	X	H	HI-Z	
H	L	H	H	L	X	X	HI-Z	
H	H	L						2
H	H	H	H	L	X	X	UBUS	3

NOTES:

1. BUF CMI bytes map into CMI bytes one to one.

2. Don't use.

3. Mapping as follows:

CMI Byte	Unibus Byte
0	0
1	1
2	0
3	1

4. BDP is an imprecise term. Actual mapping is a function of DP SEL and B0. UDP Table 5 lists Mapping by byte.

UDP Table 5

CMI DRIVE FOR PRTC = HLL

DP	SEL	BØ	CMI B3	CMI B2	CMI B1	CMI BØ
1	Ø					
L	L	L	UD B1	UD BØ	UD B1	UD BØ
L	L	H	UD BØ	UD B1	UD BØ	UD B1
L	H	X	BDP 1 B3	BDP 1 B2	BDP 1 B1	BDP 1 BØ
H	L	X	BDP 2 B3	BDP 1 B2	BDP 1 B1	BDP 1 BØ
H	L	X	BDP 3 B3	BDP 1 B2	PDB 1 B1	BDP 1 BØ

UDP Table 6

BUF CMI SOURCES

PRTC			SC	BUF CMI	NOTES
2	1	Ø	1Ø		
L	L	L	LL	CMI	
L	L	L	XH	HI-Z	
L	L	L	HX	HI-Z	
L	L	H	LH	HI-Z	
L	L	H	HL	HI-Z	
L	L	H	HH	HI-Z	
L	L	H	LL	HI-Z	1
L	H	X	HL	HI-Z	
L	H	X	XH	HI-Z	
L	H	X	LL	CMI	
H	L	L	HL	HI-Z	
H	L	L	XH		1
H	L	L	LL	CMI	
H	X	H	HL	RCAR	1
H	X	H	XH		1
H	X	H	LL		1
H	H	L	XX		1

NOTES:

1. Don't

UDP Table 7

BDPC CONTROLS

BDPC					
2	1	0	BDP	BAR	COMMENTS
L	L	L	NO LOAD	NO LOAD	IDLE STATE
L	L	H	BDP <-- CMI, UD <--BDP/CMI	NO LOAD	MEMORY DATA ON CMI FOR UBUS
L	H	L			SPARE
L	H	H	UD <-- BDP/CMI	NO LOAD	UD GETS DATA FROM BDP OR CMI IF DP = 0
H	L	L	BDP B0 <-- UD TWO BYTES	LOAD	GETTING EXTRA BYTE ON WRAP
H	L	H	BDP <-- UD TWO BYTES	LOAD	DATO
H	H	L	BDP <-- UD ONE BYTE	LOAD	DATOB
H	H	H		LOAD	SPARE

UDP Table 8

PRTC DESCRIPTION

PRTC							
2	1	Ø	UD	UA	BUF CMI	CMI	COMMENT
L	L	L	HI-Z	HI-Z	HI-Z	HI-Z	IDLE
L	L	H	BDP/CMI	HI-Z	HI-Z	HI-Z	UNIBUS DEVICE DOING DATI (P)
L	H	L	HI-Z	HI-Z	HI-Z	BUF CMI	PUT ADDRESS ON CMI WITH MAP ADDRESSED FROM UBUS
L	H	H	HI-Z	BAR	HI-Z	BUF CMI	LIKE ABOVE BUT MAP ADDRESSED BY BAR
H	L	L	HI-Z	HI-Z	HI-Z	BDP/UD	UNIBUS DEVICE DOING DATO (B)
H	L	H	CMI	HI-Z	RCAR	HI-Z	CPU DOING WRITE TO UBUS
H	H	L					SPARE
H	H	H	HI-Z	HI-Z	RCAR	UBUS	CPU DOING READ TO UBUS

NOTE:

This chart applies only for SC = HL.

A/B means A if DPSEL ≠ Ø.
B if DPSEL = Ø.

UDP Table 9

SC

SCI	SC0	UA	CMI	BUF CMI	USE	LEGAL WITH THESE PRTC VALUES	NOTE
H	L				IDLE	ALL	1
H	H	BAR	BUF CMI	HI-Z	READ BAR, CSR	LLL, LLH, LHX	
L	L	RCAR		CMI	WRITE MAP, CSR	LLL, LHX, HLL	
L	H	RCAR	BUF CMI	HI-Z	READ MAP,	LLL, LLH, LHX	

NOTE:

1. Blanks in the port fields mean no change.

3.29 MICRO-TRAP CHIP (UTR-DC628)

1. GENERAL DESCRIPTION:

This specification defines the detail requirements for a Micro-Trap Chip (UTR). UTR is a control chip which detects and prioritizes micro-trap conditions and generates a 4 bit micro vector identifying the highest priority micro-trap. UTR also contains several status registers which provide additional information about the conditions which caused the micro-trap.

UTR TABLE 1

Pin Identification and Gate Type

PIN #	PIN ID	COMMENTS	GATE TYPE
1	ENC UTRAP 1 L		GA1TNF
2	ENC UTRAP 2 L		GA1TNF
3	UTRAP L		GA1TTF
4	MICRO VECTOR 3 H		GA1TZH
5	PTE CHECK OR PROBE H		GA1TNF
6	MICRO VECTOR 1 H		GA1TZH
7	MICRO VECTOR Ø H		GA1TZH
8	MICRO VECTOR 2 H		GA1TZH
9	GEN DEST INH L		GA1TTN
1Ø	DO SRVC L		GA1TNF
11	MSRC XB H		GA1TNF
14	ENC UTRAP Ø L		GA1TNF
15	WCTRL HHLXXX L		GA1TNF
16	XB SELECT H		GA1TNF
17	XBØ IN USE L		GA1TNF
18	PROC INIT L		GA1TNF
19	RTUT DINH L		GA1TNF
2Ø	STATUS Ø H		GA1TNF
21	STATUS 1 H		GA1TNF
22	STATUS VALID L		GA1TNF
23	XB1 IN USE L		GA1TNF
24	B CLK L		GA1TNF
25	INHIBIT CMI H		GA1TTN
26	PREFETCH L		GA1TNF
27	PHASE 1 H		GA1TNF
28	WBUS 27 H		GA1TZF
29	WBUS 26 H		GA1TZF
3Ø	WBUS 25 H		GA1TZF
31	WBUS 24 H		GA1TZF
32	D CLK ENABLE H		GA1TNF
33	LATCHED WCTRL 2 H		GA1TNF
34	LATCHED WCTRL Ø H		GA1TNF
36	LATCHED WCTRL 1 H		GA1TNF
37	TB TAG Ø PERR H		GA1TNF
39	WRITE BUS ERROR INT L		GA1TTN
4Ø	TB TAG 1 PERR H		GA1TNF
41	TB HIT Ø H		GA1TNF
42	TB HIT 1 H		GA1TNF
43	TB PARITY ENA H		GA1TNF
44	M BIT H		GA1TNF
45	LATCHED BUS 3 H		GA1TNF
46	TB DATA PERR H		GA1TNF
47	ACV H		GA1TNF
48	ADD REG ENA L		GA1TNF

CHIP I/O SUMMARY

INPUTS	#PINS
ACV H	1
ADD REG ENA L	1
B CLK L	1
D CLK ENABLE H	1
DO SRVC L	1
ENC UTRAP <2:0> L	3
LATCHED BUS 3 H	1
LATCHED WCTRL <2:0> H	3
M BIT H	1
MSRC XB H	1
PHASE 1 H	1
PREFETCH L	1
PROC INIT L	1
PTE CHECK OR PROBE H	1
RTUT DINH L	1
STATUS <1:0> H	2
STATUS VALID L	1
TB DATA PERR H	1
TB HIT <1:0> H	2
TB PARITY ENA H	1
TB TAG <1:0> PERR H	2
WCTRL HHLXXX L	1
XB SELECT H	1
XB<1:0> IN USE L	2

BI-DIRECTIONAL LINES (TRI-STATE UNLESS OTHERWISE NOTED)

MICRO VECTOR <3:0> H	4
WBUS <27:24> H	4

OUTPUTS (TOTEM POLE UNLESS OTHERWISE NOTED)

GEN DEST INH L	1
INHIBIT CMI H	1
UTRAP L	1
WRITE BUS ERROR INT L	1

2. Performance:

2.1 The following is a partial list of code assignments for the WCTRL Micro field:

```
00.   PSL <- (WBUS)
20.   VA <- PC + ISIZE + (WBUS)
      PC <- PC + ISIZE
21.   VA <- VA SAVE + (WBUS)
22.   VA <- VA + 4
23.   MDR <- (WBUS)
24.   PC <- (WBUS)
```

25.	VA <- (WBUS)
26.	MBUS <- WDR
27.	MDR <- 0
28.	TB DATA <- (WBUS)
29.	TB VALID BIT <- 0 VA <- (WBUS)
2A.	WDR <- (WBUS) UNROTATED
2B.	MDR <- IR, ZERO EXTENDED
2C.	PC <- PC + (WBUS)
2D.	CACHE VALID BIT <- 0 VA <- (WBUS)
2E.	WDR <- (WBUS)
2F.	MDR <- OSR, ZERO EXTENDED
30.	STATUS/CONTROL REGISTER <- WBUS<27:24>
31.	PREVIOUS MODE REGISTER <- WBUS<23:22>
32.	WBUS<27:24> <- STATUS/CONTROL REGISTER
33.	BUS GRANT
34.	STATUS/CONTROL ADDRESS REGISTER <- WBUS<27:24>
35.	IS/CURRENT MODE REGISTER <- WBUS<26:24>
37.	REI CHECK
38.	ASTLVL REGISTER <- WBUS<26:24>
39.	(RESERVED)
3A.	WBUS<26:24> <- ASTLVL REGISTER
3B.	(RESERVED)
3C.	HIGHEST SOFTWARE IPR REGISTER <- WBUS<20:16>
3D.	IPL REGISTER <- WBUS<20:16>
3E.	NOP
3F.	WBUS<20:16> <- IPL OF HIGHEST IPR

2.2 WCTRL Functions are decoded from "LATCHED WCTRL <2:0> H" and "WCTRL HHLXXX L" which is TRUE (LOW) if the 3 most significant bits of the 6 bit "LATCHED WCTRL <5:0> H" field are:

"LATCHED WCTRL 5 H" = 1
"LATCHED WCTRL 4 H" = 1
"LATCHED WCTRL 3 H" = 0

2.3 INTERNAL SIGNAL DEFINITIONS

*"BUS GRANT DEC H" is TRUE if: (WCTRL Decode)

BUS GRANT

*"ENA UVCTR HI H" is TRUE if:

("UTRAP H" + "PTE CHECK OR PROBE H")

*"ENA UVCTR H" is TRUE if:

"UTRAP H" & "DO SRVC L"

*"UTRAP H" is TRUE if:

```
    "PHASE 1 L" & "UTRAP PEND H"

*"UTRAP PEND H" is TRUE if:

    "ENC UTRAP H" + "TB TAG ERROR H" + "TB MISS H" +
    "ACV PERR DEL H" + "BUS ERR UTRAP H" + "XB1 UTRAP H"
    + "XB0 UTRAP H" + "DO SRVC H" + "LATCHED UTRAP H"

*"ENC UTRAP H" is TRUE if:

    ("ENC UTRAP 2 H" & "ENC UTRAP 1 H") +

    "BUS UTRAP ENA H" &
    ("ENC UTRAP 2 H" + "ENC UTRAP 1 H" + "ENC UTRAP 0 H")
*"BUS UTRAP ENA H" is TRUE if:

    "ADD REG ENA H" & "RTUT DINH L" &
    "PREFETCH L" & "BUS GRANT DEC L"

*"TB TAG ERROR H" is TRUE if:

    "BUS UTRAP ENA H" &
    [("HIT 1 H" & "HIT 0 H") + "TAG 1 PERR H" + TAG 0
    PERR H"]

*"TAG 1 PERR H" is the HIGH TRUE output of a latch which
is enabled during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:

    "TB TAG 1 PERR H" & "TB PARITY ENA H"

*"FUNC LATCH ENA H" is TRUE if:

    "ADD REG ENA H" & "ADD REG ENA DEL L"

*"TAG 0 PERR H" is the HIGH TRUE output of a latch which
    is enabled during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:

    "TB TAG 0 PERR H" & "TB PARITY ENA H"

*"DATA PERR H" is the HIGH TRUE output of a latch which
    is enabled during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:
```

"TB DATA PERR H" & "TB PARITY ENA H"

*"HIT 1 H" is the HIGH TRUE output of a latch which is enabled during:

"FUNC LATCH ENA H"

LATCH INPUT IS:

"TB HIT 1 H" + "TB PARITY ENA L"

*"HIT 0 H" is the HIGH TRUE output of a latch which is enabled during:

"FUNC LATCH ENA H"

LATCH INPUT IS:

"TB HIT 0 H" & "TB PARITY ENA H"

*"AC VIO H" is the HIGH TRUE output of a latch which is enabled during:

"FUNC LATCH ENA H"

LATCH INPUT IS:

"ACV H" & "TB PARITY ENA H"

*"TB MISS H" is TRUE if:

"BUS UTRAP ENA H" &
[("LATCHED BUS 3 H" & "M BIT L" & "TB PARITY ENA H")
+
("HIT 1 L" & "HIT 0 L")]

*"XB1 UTRAP H" is TRUE if:

("XB1 UTRAP PEND H" & "XB1 IN USE H") &
("XB SELECT L" + "XB0 UTRAP PEND L")

*"XB0 UTRAP H" is TRUE if:

("XB0 UTRAP PEND H" & "XB0 IN USE H") &
("XB SELECT H" + "XB1 UTRAP PEND L")

*"XB1 UTRAP PEND H" is TRUE if:

"XB1 ACV H" + "XB1 STATUS 1 L" + "XB1 UB UNALIGNED H"
+
"XB1 TAG 1 PERR H" + "XB1 TAG 0 PERR H" + "XB1 DATA PERR H" +
"XB1 MULT HIT H" + "XB1 TB MISS H"

```
*"XB1 MULT HIT H" is TRUE if:

    "XB1 TB HIT 1 H" & "XB1 TB HIT 0 H"

*"XB1 TB MISS H" is TRUE if:

    "XB1 TB HIT 1 L" & "XB1 TB HIT 0 L"

*"XB0 UTRAP PEND H" is TRUE if:

    "XB0 ACV H" + "XB0 STATUS 1 L" + "XB0 UB UNALIGNED H"
    +
    "XB0 TAG 1 PERR H" + "XB0 TAG 0 PERR H" + "XB0 DATA
    PERR H" +
    "XB0 MULT HIT H" + "XB0 TB MISS H"

*"XB0 MULT HIT H" is TRUE if:

    "XB0 TB HIT 1 H" & "XB0 TB HIT 0 H"

*"XB0 TB MISS H" is TRUE if:

    "XB0 TB HIT 1 L" & "XB0 TB HIT 0 L"

*"LATCHED UB UNAL H" is the HIGH TRUE output of a latch
    which is enabled during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:

    "ENC UTRAP 2 H" & "ENC UTRAP 1 L" & "ENC UTRAP 0 L" &
    ["TB PARITY ENA L" +
    ("TB HIT 1 H" .XOR. "TB HIT 0 H") & "ACV L" &
    "TB TAG 1 PERR L" & "TB TAG 0 PERR L" & "TB DATA PERR
    L"]

*"PREFETCH CYC H" is the HIGH TRUE output of a latch
    which is                              enabled during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:

    "PREFETCH H"

*"READ CYC H" is the HIGH TRUE output of a latch which
    is enabled                                    during:

    "FUNC LATCH ENA H"

LATCH INPUT IS:

    "LATCHED BUS 3 L" + "PREFETCH H"
```

*"XB1 ERR ENA H" is TRUE if:

"PREFETCH CYC H" & "STATUS VAL H" &
"XB SELECT H"

*"XB0 ERR ENA H" is TRUE if:

"PREFETCH CYC H" & "STATUS VAL H" &
"XB SELECT L"

*"PHASE 2 CLK H" is TRUE if:

"PHASE 1 L" & "B CLK H"

*"EXAM TB ERR H" is TRUE if:

"ADD REG ENA DEL H" & "BUS UTRAP ENA H" & "B CLK H"

*"SET NON EXIST H" is TRUE if:

"PHASE 2 CLK H" &
[("XB1 STATUS 1 L" & "XB1 STATUS 0 L" & "XB1 UTRAP H") +
("XB0 STATUS 1 L" & "XB0 STATUS 0 L" & "XB0 UTRAP H")] +

("STAT 1 L" & "STAT 0 L" & "STATUS VAL H")

*"STAT 1 H" is the HIGH TRUE output of a latch which is enabled during:

"STATUS VAL H"

LATCH INPUT IS:

"STATUS 1 H"

*"STAT 0 H" is the HIGH TRUE output of a latch which is enabled during:

"STATUS VAL H"

LATCH INPUT IS:

"STATUS 0 H"

*"SET UNCORR H" is TRUE if:

"PHASE 2 CLK H" &
[("XB1 STATUS 1 L" & "XB1 STATUS 0 H" & "XB1 UTRAP H") +
("XB0 STATUS 1 L" & "XB0 STATUS 0 H" & "XB0 UTRAP H")] +

```
        ("STAT 1 L" & "STAT 0 H" & "STATUS VAL H")

*"SET CORR DATA H" is TRUE if:

        "PHASE 2 CLK H" &
        [("XB1 STATUS 1 H" & "XB1 STATUS 0 L" & "XB1 UTRAP
        H") +
        ("XB0 STATUS 1 H" & "XB0 STATUS 0 L" & "XB0 UTRAP
        H")] +

        ("STAT 1 H" & "STAT 0 L" & "STATUS VAL H")

    *"SET TAG 1 PERR H" is TRUE if:

        "PHASE 2 CLK H" &
        [("XB1 UTRAP H" & "XB1 TAG 1 PERR H") +
        ("XB0 UTRAP H" & "XB0 TAG 1 PERR H")] +

        ("TAG 1 PERR H" & "EXAM TB ERR H")

    *"SET TAG 0 PERR H" is TRUE if:

        "PHASE 2 CLK H" &
        [("XB1 UTRAP H" & "XB1 TAG 0 PERR H") +
        ("XB0 UTRAP H" & "XB0 TAG 0 PERR H")] +

        ("TAG 0 PERR H" & "EXAM TB ERR H")

    *"SET DATA 1 PERR H" is TRUE if:

        "PHASE 2 CLK H" &
        [("XB1 UTRAP H" & "XB1 DATA PERR H" & "XB1 TB HIT 1
        H") +
        ("XB0 UTRAP H" & "XB0 DATA PERR H" & "XB0 TB HIT 1
        H")] +

        ("DATA PERR H" & "HIT 1 H" & "EXAM TB ERR H")

    *"SET DATA 0 PERR H" is TRUE if:

        "PHASE 2 CLK H" &
        [("XB1 UTRAP H" & "XB1 DATA PERR H" & "XB1 TB HIT 0
        H") +
        ("XB0 UTRAP H" & "XB0 DATA PERR H" & "XB0 TB HIT 0
        H")] +

        ("DATA PERR H" & "HIT 0 H" & "EXAM TB ERR H")

    *"ENA WBUS H" is TRUE if:

        "PHASE 1 L" & "SC ADD 3 H" &
        WBUS <27:24> <- STATUS/CONTROL REGISTER (WCTRL
        Decode)
```

*"WRITE ERR SET H" is the HIGH TRUE output of an RS FLIP-FLOP which is set by:

"WRITE BUS ERR H" & "PHASE 1 L" & "B CLK H"

and reset by:

"WRITE BUS ERR L" & "B CLK H"

2.4 **Flip-Flop Definitions:** All flip-flops except "ADD REG ENA DEL H" are clocked on the RISING EDGE of "B CLK L".

#"ADD REG ENA DEL H" - D FLIP-FLOP (FALLING edge of "B CLK L") D INPUT:

"ADD REG ENA H" & ("PREFETCH H" + "PHASE 1 L")

The flop is DC CLEARED when "ADD REG ENA H" is FALSE.

#"STATUS VAL H" - D FLIP-FLOP
D INPUT:

"STATUS VALID H"

The flop is DC CLEARED when "STATUS VALID H" is FALSE.

#"BUS ERROR SET H" - D FLIP-FLOP
D INPUT:

"BUS ERROR BIT H"

The flop is DC CLEARED when "BUS ERROR BIT H" is FALSE.

#"BUS ERR UTRAP H" - D FLIP-FLOP
D INPUT:

("READ CYC H" & "PREFETCH CYC L" &
"STAT 1 L" & "STATUS VAL H") +

"BUS ERR UTRAP H"

The flop is DC CLEARED if:

"PROC INIT H" +
("LATCHED UTRAP H" & "PHASE 1 H" & "B CLK L")

#"LATCHED UTRAP H" - D FLIP-FLOP D INPUT:

"UTRAP H" & "DO SRVC L" & "PHASE 1 L"

#"ACV PERR DEL H" - D FLIP-FLOP
D INPUT:

"ADD REG ENA DEL H" & ("DATA PERR H" + "AC VIO H")

The flop is DC CLEARED when "ADD REG ENKA DEL H" is FALSE.

```
#"WRITE BUS ERR H" - D FLIP-FLOP
D INPUT:

     "READ CYC L" & "STATUS VAL H" & "STAT 1 L"
```

The flop is DC CLEARED if:

```
     "PROC INIT H" +
     ("WRITE ERR SET H" & "PHASE 1 H")
```

2.5 **S/C Address Register:** The S/C ADDRESS REGISTER is a 4 bit latch which is enabled during:

```
     "B CLK H" & "D CLK ENABLE H" &
     STATUS/CONTROL ADDRESS REGISTER <- WBUS <27:24> (WCTRL Decode)
```

```
LATCH Inputs are:  "WBUS <27:24> H"
LATCH Outputs are: "SC ADD <3:0> H"
```

2.6 **Status/Control Registers:**

2.6.1 **Error Summary Register:** The ERROR SUMMARY REGISTER is a 4 bit latch which is enabled during:

```
     "B CLK H" & "D CLK ENABLE H" &
     "SC ADD 3 H" & "SC ADD 2 L" & "SC ADD 1 L" & "SC ADD 0 L" &
     STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)
```

```
LATCH Inputs are:  "WBUS <27:24> H"
LATCH Outputs are:

     Bit <3>: "BUS ERROR BIT H"
     Bit <2>: "TB ERROR BIT H"
     Bit <1>: "UB UNAL BIT H"
     Bit <0>: "XB MACH CHK BIT H"
```

In addition, the individual bits may be ASYNCHRONOUSLY PRESET and/or CLEARED as follows:

Bit <3> PRESET IF:

```
     "SET NON EXIST H" + "SET UNCORR H" + "SET CORR DATA H"
```

Bit <2> PRESET IF:

```
     "PHASE 2 CLK H" &
     [("XB1 TB HIT 1 H" & "XB1 TB HIT 0 H" & "XB1 UTRAP H") +
     ("XB0 TB HIT 1 H" & "XB0 TB HIT 0 H" & "XB0 UTRAP H")] +

     ("HIT 1 H" & "HIT 0 H" & "EXAM TB ERR H")
```

Bit <1> PRESET IF:

"PHASE 2 CLK H" &
[("XB1 UB UNALIGNED H" & "XB1 UTRAP H") +
("XB0 UB UNALIGNED H" & "XB0 UTRAP H")] +

("LATCHED UB UNAL H" & "EXAM TB ERR H")

Bit <0> PRESET IF:

"PHASE 2 CLK H" &

ONE OF THE FOLLOWING MICRO-TRAPS IS DETECTED AND IS
THE HIGHEST PRIORITY: (SEE LIST, PAGE 17, HEREIN).

3,4,19,20

Bit <0> CLEARED IF:

"PHASE 2 CLK H" &

ONE OF THE FOLLOWING MICRO-TRAPS IS DETECTED AND IS
THE HIGHEST PRIORITY: (SEE Section 2.7 HEREIN).

5,6,9

All 4 Bits are ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

2.6.2 **BUS Error Register:** The BUS ERROR REGISTER is a 4 bit register constructed of RS FLIP-FLOPS.

REGISTER OUTPUTS ARE.

Bit <3>: "NON EXIST MEM H"
Bit <2>: "UNCORR DATA H"
Bit <1>: "LOST ERROR H"
Bit <0>: "CORR DATA H"

Bit <3> is set by,:

"SET NON EXIST H"

Bit <2> is set by:

"SET UNCORR H"

Bit <1> is set by:

"BUS ERROR SET H" &
("SET NON EXIST H" + "SET UNCORR H" + "SET CORR DATA H")

Bit <0> is set by:

"SET CORR DATA H"

All 4 bits are ASYNCHRONOUSLY RESET when "BUS ERROR BIT H" is FALSE.

2.6.3 **Parity Error Register:** The PARITY ERROR REGISTER is a 4 bit register constructed of RS FLIP-FLOPS.

REGISTER OUTPUTS ARE:

Bit <3>: "TAG 1 PERR BIT H"
Bit <2>: "TAG 0 PERR BIT H"
Bit <1>: "DATA 1 PERR BIT H"
Bit <0>: "DATA 0 PERR BIT H"

Bit <3> is set by:

"SET TAG 1 PERR H"

Bit <2> is set by:

"SET TAG 0 PERR H"

Bit <1> is set by:

"SET DATA 1 PERR H"

Bit <0> is set by:

"SET DATA 0 PERR H"

All 4 bits are ASYNCHRONOUSLY RESET when "TB ERROR ERROR BIT H" is FALSE.

2.6.4 **Last Reference Hit Register:** The LAST REFERENCE HIT REGISTER is a one bit latch which is enabled during:

"EXAM TB ERR H"

LATCH INPUT IS: ("HIT 1 H" .xor. "HIT 0 H")
LATCH OUTPUT IS: "LAST REF HIT H"

The latch is ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

2.6.5 **Reference Cache Only Register:** The REFERENCE CACHE ONLY REGISTER is a one bit latch which is enabled during:

"B CLK H" & "D CLK ENABLE H" &
"SC ADD 3 H" & "SC ADD 2 H" & "SC ADD 1 H" & "SC ADD 0 L" &
STATUS/CONTROL REGISTER <- WBUS <27:24> (WCTRL Decode)

LATCH INPUT IS: "WBUS 24 H"
LATCH OUTPUT IS: "DISABLE CMI REG H"

The latch is ASYNCHRONOUSLY CLEARED when "PROC INIT H" is TRUE.

2.6.6 **XB1 Status Register:** The XB1 STATUS REGISTER is a 9 bit latch which is enabled during:

"XB1 ERR ENA H"
LATCH INPUTS -> LATCH OUTPUTS ARE:

"HIT 1 H"	->	"XB1 TB HIT 1 H"
"HIT 0 H"	->	"XB1 TB HIT 0 H"
"STAT 1 H"	->	"XB1 STATUS 1 H"
"STAT 0 H"	->	"XB1 STATUS 0 H"
"AC VIO H"	->	"XB1 ACV H"
"TAG 1 PERR H"	->	"XB1 TAG 1 PERR H"
"TAG 0 PERR H"	->	"XB1 TAG 0 PERR H"
"DATA PERR H"	->	"XB1 DATA PERR H"
"LATCHED UB UNAL H"	->	"XB1 UB UNALIGNED H"

2.6.7 **XB0 Status Register:** The XB0 STATUS REGISTER is a 9 bit latch which is enabled during:

"XB0 ERR ENA H"

LATCH INPUTS -> LATCH OUTPUTS ARE:

"HIT 1 H"	->	"XB0 TB HIT 1 H"
"HIT 0 H"	->	"XB0 TB HIT 0 H"
"STAT 1 H"	->	"XB0 STATUS 1 H"
"STAT 0 H"	->	"XB0 STATUS 0 H"
"AC VIO H"	->	"XB0 ACV H"
"TAG 1 PERR H"	->	"XB0 TAG 1 PERR H"
"TAG 0 PERR H"	->	"XB0 TAG 0 PERR H"
"DATA PERR H"	->	"XB0 DATA PERR H"
"LATCHED UB UNAL H"	->	"XB0 UB UNALIGNED H"

2.7 **Micro Trap Conditions:** The following MICRO-TRAPS are detected and prioritized in the order shown: See Para. 2.8 for Functions:

(HIGHEST) 1. CONTROL STORE PARITY ERROR
2. FPA RESERVED OPERAND
3. MSRC XB TB ERROR
4. MSRC XB BUS ERROR
5. BUS ERROR
6. UNALIGNED UNIBUS DATA
7. MSRC XB TB MISS
8. MSRC XB ACV
9. TB ERROR
10. TB MISS, READ
11. TB MISS, WRITE
12. ACV, READ
13. ACV, WRITE
14. WRITE CROSSING PAGE BOUNDARY

15. WRITE UNLOCK CROSSING PAGE BOUNDARY
16. UNALIGNED DATA, READ
17. UNALIGNED DATA, WRITE
18. UNALIGNED DATA, WRITE UNLOCK
19. BUT XB TB ERROR
20. BUT XB BUS ERROR
21. BUT XB TB MISS
22. BUT XB ACV

Micro Vector assignments shall be in accordance with UTR Table 2.

UTR TABLE 2

MICRO-	*UVECTOR			
TRAP #	3 H*	2 H*	1 H*	0 H*
1	L	L	L	L
2	H	H	L	L
3	H	L	L	L
4	H	L	L	L

UTR TABLE 2 (Cont)

MICRO-	"UVECTOR			
TRAP #	3 H"	2 H"	1 H"	0 H"
5	H	L	L	L
6	H	L	L	L
7	L	L	H	L
8	L	L	H	H
9	H	L	L	L
10	H	L	H	L
11	H	L	H	H
12	H	H	H	L
13	H	H	H	H
14	L	H	H	H
15	L	H	H	L
16	L	L	L	H
17	L	H	L	H
18	L	H	L	L
19	H	L	L	L
20	H	L	L	L
21	H	L	L	H
22	H	H	L	H

2.8 **Micro-Trap Functions:** As specified herein.

CONTROL STORE PARITY ERROR Micro-Trap condition exists if:

"ENC UTRAP 2 H" & "ENC UTRAP 1 H" & "ENC UTRAP 0 H"

FPA RESERVED OPERAND Micro-Trap condition exists if:

```
    "ENC UTRAP 2 H" & "ENC UTRAP 1 H" & "ENC UTRAP 0 L"
```

MSRC XB TB ERROR Micro-Trap condition exists if:

```
    "MSRC XB H" &
    ["XB1 UTRAP H" &
    ("XB1 TAG 1 PERR H" + "XB1 TAG 0 PERR H" +
    "XB1 DATA PERR H" + "XB1 MULT HIT H") +

    "XB0 UTRAP H" &
    ("XB0 TAG 1 PERR H" + "XB0 TAG 0 PERR H" +
    "XB0 DATA PERR H" + "XB0 MULT HIT H")]
```

MSRC XB BUS ERROR Micro-Trap condition exists if:

```
    "MSRC XB H" &
    [("XB1 UTRAP H" & "XB1 STATUS 1 L") +

    ("XB0 UTRAP H" & "XB0 STATUS 1 L")]
```

BUS ERROR Micro-Trap condition exists if:

```
    "BUS ERR UTRAP H"
```

UNALIGNED UNIBUS DATA Micro-Trap condition exists if:

```
    ("XB1 UB UNALIGNED H" & "XB1 UTRAP H") +
    ("XB0 UB UNALIGNED H" & "XB0 UTRAP H") +

    ("LATCHED UB UNAL H" & "BUS UTRAP ENA H")
```

MSRC XB TB MISS Micro-Trap condition exists if:

```
    "MSRC XB H" &
    [("XB1 UTRAP H" & "XB1 TB MISS H") +

    ("XB0 UTRAP H" & "XB0 TB MISS H")]
```

MSRC XB ACV Micro-Trap condition exists if:

```
    "MSRC XB H" &
    [("XB1 UTRAP H" & "XB1 ACV H") +

    ("XB0 UTRAP H" & "XB0 ACV H")]
```

TB ERROR Micro-Trap condition exists if:

```
    "BUS UTRAP ENA H" &
    ["TAG 1 PERR H" + "TAG 0 PERR H" + "DATA PERR H" +
    ("HIT 1 H" & "HIT 0 H")]
```

TB MISS, READ Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" & "LATCHED BUS 3 L" &
"HIT 1 L" & "HIT Ø L"
```

TB MISS, WRITE Micro-Trap condition exists if:

```
("BUS UTRAP ENA H" & "LATCHED BUS 3 H") &
[("HIT 1 L" & "HIT Ø L") +

("M BIT L" & "TB PARITY ENA H")]
```

ACV, READ Micro-Trap condition exists if:

```
"ACV PERR DEL H" & "AC VIO H" & "LATCHED BUS 3 L"
```

ACV, WRITE Micro-Trap condition exists if:

```
"ACV PERR DEL H" & "AC VIO H" & "LATCHED BUS 3 H"
```

WRITE CROSSING PAGE BOUNDARY Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" &
"ENC UTRAP 2 L" & "ENC UTRAP 1 H" & "ENC UTRAP Ø L"
```

WRITE UNLOCK CROSSING PAGE BOUNDARY Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" &
"ENC UTRAP 2 L" & "ENC UTRAP 1 H" & "ENC UTRAP Ø H"
```

UNALIGNED DATA, READ Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" & "LATCHED BUS 3 L"
"ENC UTRAP 2 L" & "ENC UTRAP 1 L" & "ENC UTRAP Ø H"
```

UNALIGNED DATA, WRITE Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" & "LATCHED BUS 3 H"
"ENC UTRAP 2 L" & "ENC UTRAP 1 L" & "ENC UTRAP Ø H"
```

UNALIGNED DATA, WRITE UNLOCK Micro-Trap condition exists if:

```
"BUS UTRAP ENA H" & "LATCHED BUS 3 H"
"ENC UTRAP 2 H" & "ENC UTRAP 1 L" & "ENC UTRAP Ø H"
```

BUT XB TB ERROR Micro-Trap condition exists if:

```
"MSRC XB L" &
["XB1 UTRAP H" &
("XB1 TAG 1 PERR H" + "XB1 TAG Ø PERR H" +
"XB1 DATA PERR H" + "XB1 MULT HIT H") +
"XB0 UTRAP H" &
("XB0 TAG 1 PERR H" + "XB0 TAG 0 PERR H" +
"XB0 DATA PERR H" + "XB0 MULT HIT H")]
```

BUT XB BUS ERROR Micro-Trap condition exists if:

```
"MSRC XB L" &
[("XB1 UTRAP H" & "XB1 STATUS 1 L") +

("XB0 UTRAP H" & "XB0 STATUS 1 L")]
```

BUT XB TB MISS Micro-Trap condition exists if:

```
"MSRC XB L" &
[("XB1 UTRAP H" & "XB1 TB MISS H") +

("XB0 UTRAP H" & "XB0 TB MISS H")]
```

BUT XB ACV Micro-Trap condition exists if:

```
"MSRC XB L" &
[("XB1 UTRAP H" & "XB1 ACV H") +

("XB0 UTRAP H" & "XB0 ACV H")]
```

UVECTOR REGISTER

The UVECTOR REGISTER is a 4 bit latch which is enabled during:

```
"LATCHED UTRAP L" & "UTRAP H"
```

The outputs of the latch, "UVECTOR <3:0> H", contain the Micro Vector of the HIGHEST PRIORITY MICRO-TRAP detected while the latch is enabled.

3.3.8 Chip Outputs:

>>> "MICRO VECTOR 3 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

```
"ENA UVCTR HI H"
```

and is TRUE (HIGH) if:

```
"DO SRVC H" +

("UTRAP L" & "ENC UTRAP 0 L" &
"MICRO VECTOR 1 H" & "MICRO VECTOR 0 H") +

("UTRAP H" & "UVECTOR 3 H")
```

>>> "MICRO VECTOR 2 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

```
"ENA UVCTR HI H" & "DO SRVC L"
```

```
and is TRUE (HIGH) if:

          ("UTRAP L" & "M BIT H" &
          "MICRO VECTOR 1 H" & "MICRO VECTOR 0 H") +

          ("UTRAP H" & "UVECTOR 2 H")

>>>   "MICRO VECTOR 1 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

          "ENA UVCTR H"

and is TRUE (HIGH) if:

          "UVECTOR 1 H"

>>>   "MICRO VECTOR 0 H" (TRI-STATE, BI-DIRECTIONAL) is enabled if:

          "ENA UVCTR H"

and is TRUE (HIGH) if:

          "UVECTOR 0 H"

>>>   "WBUS 27 H" (TRI-STATE, BI-DIRECTIONAL;  is enabled if:

          "ENA WBUS H"

and is TRUE (HIGH) if:

          ("TAG 1 PERR BIT H" & "SC ADD 2 H" & "SC ADD 0 H") +
          ("NON EXIST MEM H" & "SC ADD 2 L" & "SC ADD 0 H") +
          ("BUS ERROR BIT H" & "SC ADD 2 L" & "SC ADD 0 L")

>>>   "WBUS 26 H" (TRI-STATE, BI-DIRECTIONAL)  is enabled if:

          "ENA WBUS H"

and is TRUE (HIGH) if:

          ("TAG 0 PERR BIT H" & "SC ADD 2 H" & "SC ADD 0 H") +
          ("UNCORR DATA H" & "SC ADD 2 L" & "SC ADD 0 H") +
          ("TB ERROR BIT H" & "SC ADD 2 L" & "SC ADD 0 L")

>>>   "WBUS 25 H" (TRI-STATE, BI-DIRECTIONAL)  is enabled if:

          "ENA WBUS H"

and is TRUE (HIGH) if:

          ("DATA 1 PERR BIT H" & "SC ADD 2 H" & "SC ADD 0 H") +
          ("LOST ERROR H" & "SC ADD 2 L" & "SC ADD 0 H") +
          ("UB UNAL BIT H" & "SC ADD 2 L" & "SC ADD 0 L")
```

```
>>>   "WBUS 24 H" (TRI-STATE, BI-DIRECTIONAL)  is enabled if:

         "ENA WBUS H"

  and is TRUE (HIGH) if:

         ("DATA 0 PERR BIT H" & "SC ADD 2 H" & "SC ADD 0 H") +
         ("CORR DATA H" & "SC ADD 2 L" & "SC ADD 0 H") +
         ("XB MACH CHK BIT H" & "SC ADD 2 L" & "SC ADD 0 L") +
         ("DISABLE CMI REG H" & "SC ADD 2 H" & "SC ADD 1 H") +
        ("LAST REF HIT H" & "SC ADD 2 H" & "SC ADD 1 L" & "SC ADD 0 L")

>>>   "GEN DEST INH L" is TRUE (LOW) if:

        "UTRAP PEND H" & ["DO SRVC H" +
        (UNALIGNED DATA, WRITE UNLOCK and/or any HIGHER PRIORITY
        Micro-Trap conditions exist)]

>>>   "INHIBIT CMI H" is TRUE (HIGH) if:

        "DISABLE CMI REG H" + ("RTUT DINH H" & "PREFETCH L") +

        "PREFETCH H" &
        ["AC VIO H" +
        "TAG 1 PERR H" + "TAG 0 PERR H" + "DATA PERR H" +
        ("HIT 1 H" & "HIT 0 H") +
        ("HIT 1 L" & "HIT 0 L")]

>>>   "UTRAP L" is TRUE (LOW) if:

        "UTRAP H"

>>>   "WRITE BUS ERROR INT L" is TRUE (LOW) if:

        "WRITE BUS ERR H"
```

```
                              ---------------
          ENC UTRAP 1 L --->!01           48!<--- ADD REG ENA L
          ENC UTRAP 2 L --->!02           47!<--- ACV H
                UTRAP L <---!03           46!<--- TB DATA PERR H
     MICRO VECTOR 3 H <---!04           45!<--- LATCHED BUS 3 H
PTE CHECK OR PROBE H --->!05           44!<--- M BIT H
     MICRO VECTOR 1 H <-->!06           43!<--- TB PARITY ENA H
     MICRO VECTOR 0 H <-->!07           42!<--- TB HIT 1 H
     MICRO VECTOR 2 H <---!08           41!<--- TB HIT 0 H
        GEN DEST INH L <---!09           40!<--- TB TAG 1 PERR H
             DO SRVC L --->!10 ....... 39!---> WRITE BUS ERROR INT L
             MSRC XB H --->!11 .     . 38!----GROUND
                   VGA ----!12 . LID . 37!<--- TB TAG 0 PERR H
                   VCC ----!13 . DOWN. 36!<--- LATCHED WCTRL 1 H
          ENC UTRAP 0 L --->!14 .     . 35!----GROUND
        WCTRL HHLXXX L --->!15 ....... 34!<--- LATCHED WCTRL 0 H
           XB SELECT H --->!16           33!<--- LATCHED WCTRL 2 H
          XB0 IN USE L --->!17           32!<--- D CLK ENABLE H
           PROC INIT L --->!18           31!<--> WBUS 24 H
           RTUT DINH L --->!19           30!<--> WBUS 25 H
            STATUS 0 H --->!20           29!<--> WBUS 26 H
            STATUS 1 H --->!21           28!<--> WBUS 27 H
        STATUS VALID L --->!22           27!<--- PHASE 1 H
          XB1 IN USE L --->!23           26!<--- PREFETCH L
               B CLK L --->!24           25!---> INHIBIT CMI H
                              ---------------
```

UTR FIGURE 1

PIN CONFIGURATION DIAGRAM AND PIN IDENTIFICATION